

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.43

«До захисту допущено»

Завідувач кафедри
Ігор ПАРХОМЕЙ
(підпис)

“ ” 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Веб-сервіс для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних» _____

Виконав:

студент II курсу, групи ІК-91мп
Юраш Антон Павлович _____

Керівник:

доц. кафедри ТК ФІОТ, к. т. н., доцент,
Лісовиченко Олег Іванович _____

Консультант з нормоконтролю:

доцент, к.т.н., доц.,
Пасько Віктор Петрович _____

Рецензент:

проф. каф. ОТ, д.т.н., проф.,
Стіренко Сергій Григорович _____

Засвідчую, що у цій
магістерській дисертації немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Ігор ПАРХОМЕЙ
(підпис)
«__» _____ 2020 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Юраша Антона Павловича

(прізвище, ім'я, по батькові)

1. Тема дисертації «Веб-сервіс для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних»,

науковий керівник дисертації: к. т. н., доцент Лісовиченко Олег Іванович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «28» 10 2020 р. № 3770-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження: системи обліку навчальних даних про успішність студентів..

4. Предмет дослідження: Вебсервіс для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.

5. Перелік завдань, які потрібно розробити: 1. Огляд існуючих систем обліку навчальних даних. 2. На основі аналізу сучасних систем обліку та

контролю успішності побудувати архітектуру системи. 3. Розробка веб-сервісу для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану.

6. Орієнтовний перелік ілюстративного матеріалу: Додаток В.- схема бази даних, Додаток Г – архітектура вебсервісу, Додаток Д – опис моделей класів ORM, Додаток Ж – 6 плакатів.

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	Лісовиченко О.І., доцент	03.12.20	03.12.20
Нормоконтроль	Пасько В.П., доцент	10.12.20	10.12.20

8. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Розробка архітектури вебсервісу	02.09-08.09	
	Розробка схеми бази даних	09.09-15.09	
	Розробка Django застосунку	16.09-22.09	
	Розробка інтеграції з Google API	23.09-06.10	
	Інтеграція чатботу для розсилки сповіщень	07.10-13.10	
	Відладка та конфігурування для деплою	14.10-27.10	
	Оформлення пояснювальної записки та додатків	28.10-22.11	
	Перевірка на співпадіння		
	Захист		

Студент

А.П. Юраш

(підпис)

(ініціали, прізвище)

Науковий керівник дисертації

О.І. Лісовиченко

(підпис)

(ініціали, прізвище)

АНОТАЦІЯ

У представленій роботі розглянуто розробку веб-сервісу для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.

Було проведено аналіз існуючих систем обліку та контролю виконання навчального плану, виявлено їх основні недоліки і переваги. Розроблено архітектуру веб сервісу та схему бази даних, розроблено Django застосунок. Виконано інтеграцію із Google API та чатботом для розсилки сповіщень.

Результатом виконаної роботи є розроблений застосунок, що є платформою, що дозволяє викладачам створювати таблицю-журнал груп із певної дисципліни, керувати нею, здійснювати обмін інформацією (повідомлення про оновлення даних чи інші події), студентам дає можливість отримувати новини навчального закладу й контролювати власні досягнення при вивченні предмета.

Ключові слова: вебсервіс, Google API, інтеграція, python, PostgreSQL, Django, контроль виконання навчального плану.

Розмір пояснювальної записки – 68 сторінок, 34 ілюстрації, 22 таблиці, 5 додатків.

ABSTRACT

The presented work considers the development of a web service for the implementation of the management of an interactive curriculum accounting and control system with the structuring and storing data service integration.

The analysis of the existing curriculum accounting and control systems implementation was carried out, their main disadvantages and advantages were revealed. Developed web service architecture and database schema, developed Django application. Integrated with Google API and chatbot to send notifications.

The result of this work is a developed application that is a platform that allows teachers to create a table-journal of groups in a particular discipline, manage it, exchange information (notifications about data updates or other events), gives students the opportunity to receive news from the school and monitor their own achievements when studying the subject.

Keywords: web service, Google API, integration, python, PostgreSQL, Django, curriculum monitoring.

The size of the explanatory note is 68 pages, 34 illustrations, 22 tables, 5 appendices.

Пояснювальна записка до магістерської дисертації

на тему: Веб-сервіс для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП	9
1. СИСТЕМИ ОБЛІКУ, КОНТРОЛЮ ВИКОНАННЯ НАВЧАЛЬНОГО ПЛАНУ	12
1.1. Рейтингова система оцінювання. Система оцінювання ECTS	12
1.2. Існуючі системи обліку успішності студентів	17
1.2.1. Застосунки для локального використання.....	18
1.2.2. Вебсервіси для обліку успішності студентів	21
1.3. Задачі, котрі мають бути реалізовані у вебсервісі для обліку успішності студентів	23
Висновки до розділу	24
2. ВИБІР ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБСЕРВІСУ ДЛЯ ОБЛІКУ УСПІШНОСТІ.....	25
2.1. Мова програмування Python	25
2.2. Фреймворк для веброзробки на мові Python. Django.....	27
2.3. База даних PostgreSQL.....	32
2.4. Python обгортка навколо API Google Sheets	34
2.5. Месенджер для інтеграції сповіщень засобом чат-бота – Telegram.....	34
3. РОЗРОБКА ВЕБСЕРВІСУ ДЛЯ ОБЛІКУ УСПІШНОСТІ.....	36
3.2. Інтеграція табличного сервісу. Локальне завантаження списків груп	39
3.3. Розробка інтерфейсів вебсервісу для обліку успішності студентів	40
3.4. Розробка алгоритму роботи чат-бота для сповіщень користувачів	43
3.5. Використання застосунку	44
Висновки до розділу	46
4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ	47
4.1. Опис ідеї проєкту	47
4.2. Технологічний аудит ідеї проєкту	49
4.3. Аналіз ринкових можливостей запуску стартап-проєкту	49
4.4. Розроблення ринкової стратегії проєкту.....	55
4.5. Розроблення маркетингової програми стартап-проєкту	57
Висновки до розділу	59
ВИСНОВКИ.....	61
ПЕРЕЛІК ПОСИЛАНЬ	63
ДОДАТКИ.....	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

ВС – веб-сервіс

БД – база даних

СО – система оцінювання

МП – мова програмування

СУБД – система управління базами даних

ECTS – кредитна система, яка пропонує спосіб вимірювання та порівняння навчальних досягнень і переведення їх з одного інституту до іншого.

Python – мова програмування, яка характеризується універсальністю у використанні та якісним рівнем організації.

PostgreSQL – об'єктно-реляційна система управління базами даних (СУБД).

API – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

РСО – рейтингова система оцінювання.

ВСТУП

Проблема автоматизації сповіщення студентів про навчальні успіхи та своєчасного ведення рейтингів існує вже давно, але особливо актуальною вона стала у зв'язку з переходом на онлайн навчання у зв'язку з пандемією 2020 року. На даний момент існують лише окремі частини системи, такі як КПП-розклад, КПП-кампус, окремі бази даних деканатів. Централізована система, здатна до синхронізованої роботи зі своєчасним оновленням всіх складових елементів відсутня. Через це керування багатьма розрізненими ресурсами досить ускладнене і часто можна зіткнутися із невідповідністю інформації в різних джерелах через її застарілість або насправність якогось із елементів.

Також проблемою є сповіщення студентів і, в цілому, актуальність інформування про їх навчальні успіхи, саме тому, в даній роботі буде розглянуте рішення інтеграції чат-бота у вебсервіс, за допомогою якого студент зможе своєчасно отримувати інформацію про будь-які оцінки чи інші події, що фіксуватимуться в таблицях успішності. Такий «лог» оцінок допоможе зробити навчальний процес більш прозорим для студентів та викладачів.

У наш час робота з обліковими системами переважно переводиться в електронний формат. Однак для великих розподілених систем на зразок системи контролю та обліку виконання навчального плану студентами Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» все ще залишається вкрай актуальним завданням створення єдиної, централізованої та зручної у використанні системи. Для створення єдиної спільної системи для студентів та викладачів пропонується використання різних сервісів для структуризації та збереження даних. Правильним рішенням буде скористатися сервісом онлайн таблиць, котрим дуже зручно користуватися викладачам для ведення електронного оцінювання.

Також під питанням, в цілому, перехід на повний електронний облік в результаті впливу пандемії COVID-19 на організаційні та навчальні процеси. Потрібні рішення не лише для ведення обліку, а і для своєчасного сповіщення студентів про будь-які зміни. Існує проблема ієрархії інформування студентів,

коли від викладацького складу інформацію отримують старости, а вони вже мають донести її до груп. Така ж проблема з кураторами груп від університету. На кожному етапі ієрархії інформація може втрачатися, що негативно позначається на організації робочих процесів.

Об'єктом дослідження є дані навчального плану та інформація про успішність студентів

Предмет дослідження – методи обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.

Актуальність. Наразі у 2020 році у зв'язку з пандемією COVID-19, коли більшість навчальних закладів перейшли повністю чи гібридно на дистанційний режим навчання, актуально, як ніколи, постало питання обліку навчальних успіхів і сповіщення студентів про стан їх навчання в онлайн режимі. Також існуючі рішення на прикладі кампусу КПП показали свою нестабільність і мінуси централізованої системи, коли під впливом падінь сервісу довелося перенести атестацію. Якщо падає єдиний університетський сервіс, то відповідно виникає проблема обліку на локальному рівні, факультету чи кафедри. Отже, доречним і сучасним рішенням буде децентралізація університетських сервісів та винесення системи обліку окремо, а також реалізація системи сповіщення в окремому мікросервісі для розподілу навантажень і зменшення ризиків.

Важливим аргументом на користь рішення з інтеграцією сторонніх табличних сервісів є їх перевірена стабільність та наявність існуючого функціоналу для використання формул для підрахунків, наприклад середніх балів чи їх сум по Болонській системі. На прикладі табличного сервісу Google також маємо готове рішення розмежування доступів для студентів та викладачів (на редагування і читання) і відкрите зручне API для інтеграції.

Також актуальність даного рішення підтверджує поточна методика викладачів у веденні рейтингових балів (на листочку в таблиці, в таблиці Microsoft Excel або в табличному сервісі Google). Якщо відкинути рішення з листочком, котре є ненадійним і неефективним під час онлайн навчання в умовах карантину, залишаться таблиці Excel та гугл. Останні є кращим рішенням, адже

мають зручне й офіційне API для інтеграції, а також одразу знаходяться в онлайні, і в реальному часі їх можна переглядати з браузера без потреби в завантаженні файлу.

Новизна полягає в розробці ПЗ, котре дасть змогу зручно для викладачів і студентів вести облік та отримувати своєчасні сповіщення про події навчання. Тобто в момент виставлення чи редагування оцінки викладачем студент одразу бачить зміну, отримавши сповіщення засобом чат-бота.

1. СИСТЕМИ ОБЛІКУ, КОНТРОЛЮ ВИКОНАННЯ НАВЧАЛЬНОГО ПЛАНУ

Оскільки ми живемо у світі, де активно розвиваються сучасні технології, це спричиняє зростання запитів користувачів на програмні засоби. Застосування саморегульованих систем керування значно полегшує роботу людини, чинить позитивний вплив на розвиток суспільства в цілому. Такий прогрес має авторитет і в освітній галузі.

У наш час є багато систем, які дозволять вести облік успішності студентів. Варто зазначити, що їх реалізація допомагає при вирішенні труднощів, але ці програмні продукти часто можуть бути незручними, а то й складними у використанні.

Для початку розглянемо середовище й умови, в яких повинен працювати наш майбутній вебсервіс.

РСО – рейтингова система оцінювання, в умовах якої і передбачається робота даного програмного продукту.

Також потрібно звернути увагу на загальноприйняту систему оцінювання ECTS, котра регулює більшість параметрів, які треба врахувати при розробці даного програмного продукту. Оскільки ця система використовується в більшості європейських країн і в тому числі в Україні, розробка під її стандарти дозволить використовувати вебсервіс в тих місцях, де користуються вищезазначеною системою ECTS.

1.1. Рейтингова система оцінювання. Система оцінювання ECTS

Рейтингову систему оцінювання складають перевірка виконання конкретно поставлених завдань та навчальні досягнення студентів з певної дисципліни (накопичення балів). Використання 100-бальної шкали створює умови для прозорого оцінювання успішності студента, допомагає при формуванні рейтингу досягнень, на основі якого призначають стипендію. РСО має ряд переваг: спонукає студента до старанної підготовки під час навчання, створює умови для самостійної та систематичної роботи, є значним полегшенням

при здачі сесії, унеможливилося нечесну конкуренцію. Відповідно підвищується якісний рівень підготовки кадрів і є можливість коригування навчальної діяльності шляхом зворотного зв'язку. Інформацію про принципи функціонування рейтингової системи оцінювання повідомляють студентам на першому занятті.

Якщо навчальний предмет закінчується заліком, то використовують 100-бальну систему оцінювання, якщо ж екзаменом – то ця шкала поділяється на 2 складові (стартова й екзаменаційна). Це рекомендованих 40-60 балів за досягнення протягом вивчення дисципліни й 60-40 балів за екзамен. РСО протягом навчальної діяльності студента складають індивідуальні завдання, лабораторні роботи, звіти, оцінювання на заняттях, колоквіуми. Кожен із цих видів контролю закінчується виставленням вагових балів. Для визначення їх значень використовують наступну формулу:

$$R_K = R \frac{t_k}{\sum_i t_i} \quad (1.1)$$

де R_K — значення вагового бала з контрольного заходу.

Тобто складовими значення є: розмір шкали (R), навчальний час (запланований у програмі заздалегідь), необхідний для досягнення результатів із певної дисципліни (t_k). Уміння й навички, набуті в процесі вивчення предмета, мають контролюватись k -м контрольным заходом. Усі контрольні види оцінювання (заплановані в РСО) охоплює $\sum_i t_i$ – загальний час, протягом якого засвоюють навчальний матеріал.

У результаті число отриманих балів за контрольні види оцінювання повинно складати 100, якщо навчальна дисципліна закінчується заліком, і відповідати розміру стартової шкали ($R_c = \sum_k R_K$), якщо вкінці передбачено екзамен.

РСО має відповідати певним стандартам. Для цього розробили критерії оцінювання – рівні засвоєння навчального матеріалу. Це стосується як контрольних заходів, так і екзаменів. Для оцінки навчальних досягнень студентів

використовують дво- й чотирьох рівневу систему. До останньої входять наступні рівні: «відмінно» ($1 \dots 0,9$) R_K , «добре» ($0,89 \dots 0,75$), «задовільно» ($0,74 \dots 0,6$), «незадовільно» (0). Бал "незадовільно" виставляють, якщо студент не з'явився на екзамен чи залік, невчасно виконав завдання, був відсутній на заняттях без поважних причин. Участь в олімпіадах чи інших заходах університету також може оцінюватися додатковими балами. Сума балів за штрафи чи заохочення не має бути вищою, ніж $0,1 R_C$. Якщо вкінці вивчення предмета передбачений залік, то сума всіх балів з кредитного модуля, штрафних і заохочувальних балів становить рейтингову оцінку студента. Загальноприйнята університетська шкала балів складається з таких оцінок: «відмінно» ($95 \leq RD \leq 100$), «дуже добре» ($85 \leq RD \leq 94$), «добре» ($75 \leq RD \leq 84$), «задовільно» ($65 \leq RD \leq 74$), «достатньо» ($60 \leq RD \leq 64$), «незадовільно» ($RD < 60$) «не допущено» – студент не виконав умов для допуску.

Якщо студент протягом вивчення навчальної дисципліни набрав необхідну кількість балів ($RD \geq 60$), то отримує залік "автоматом", якщо ж менше – повинен скласти його. Щоб заохотити студентів до старанного виконання завдань протягом семестру, рейтинговий бал із кредитного модуля можна визначати як суму балів із залікової роботи та балів семестрового завдання. Ті, хто протягом семестру перейшли шкалу 60 балів, мають змогу покращити свій рейтинговий бал шляхом виконання залікових завдань.

Оцінка за контрольну роботу може бути більшою чи меншою за рейтинговий бал. Якщо більший, то виставляють бал, отриманий за контрольну. У випадку меншої оцінки за контрольне завдання, попередній рейтинговий бал може бути скасований, і тоді врахують оцінку за залікову роботу (жорсткий варіант). М'який варіант – виставлення більшої з отриманих оцінок.

Для підрахунку рейтингового бала за екзамен сумують бали за поточну успішність і бали за екзамен. Отримані результати переводять в систему оцінювання університету. Що стосується курсових робіт, то їх оцінюють окремо за двома складовими (перша – якість графічного матеріалу й пояснювальної записки, друга – якість захисту роботи). Схема розробки РСО повинна виглядати

наступним чином: тип рейтингової системи оцінювання визначають залежно від виду атестації вкінці семестру; потрібно підібрати систему видів контролю протягом вивчення навчальної дисципліни; узгодити вагові бали за кожний контрольний захід; для кожного контрольного заходу треба обрати відповідну систему оцінювання; врахувати інші елементи РСО ("м'яка" чи "жорстка" рейтингова система, наявність штрафів чи балів заохочення, особливості допуску до атестації по закінченню дисципліни). Варто зазначити, що при змішаному типі навчання РСО потрібно формувати у взаємодії із новою системою контрольних заходів.

Система ECTS пропонує залежність здійснення обліку успішності студентів від загальноприйнятої національної шкали оцінок. Окрім того, як додатковий метод, рекомендовано використовувати оцінювання ECTS grading scale, що був запропонований Європейською комісією в середині 90-х рр. у якості складової ECTS в цілому. Вище згадану систему характеризують такі ознаки:

- одиницею вимірювання трудовитрат студента, необхідних для досягнення очікуваних результатів, є ECTS-кредити; навчальний рік відповідає 60-тюм одиницям ECTS, що складає близько 1500-1800 навчальних годин. Відповідно одна залікова одиниця містить 25-30 академічних годин;
- в загальному для того, щоб отримати ступінь бакалавра, особі, яка навчається, потрібно набрати від 180 до 240 ECTS-кредитів за весь період навчання. Для отримання звання «магістр» загальна кількість кредитів повинна бути не менше 300;
- кількість кредитів має становити ціле число (як виняток допускається додавання 0.5 кредиту); за семестр загальна їх сума повинна бути 30;
- після закінчення навчальної діяльності за програмою здійснюється нарахування кредитів в цілому або ж з урахуванням її окремого структурного елемента та успішної оцінки досягнутих результатів;
- оцінювання не впливає на кількість зарахованих ECTS-одиниць із дисципліни;

- на зарахування кредитів впливають наступні чинники: самостійна робота студента, аудиторне навантаження, участь в семінарах, стажування на робочому місці, підготовка доповідей, виконання лабораторних робіт, написання рефератів, курсових;
- адміністрація навчального закладу залишає за собою право знижувати кількість набраних кредитів при умові відсутності студента на заняттях із певної дисципліни, проте в більшості відвідування занять не впливає на їх нарахування.

Оцінювання у системі ECTS має певну послідовність. Спершу потрібно встановити мінімальне значення, що свідчить про успішність у засвоєнні дисципліни й отримання кредиту. Це можна здійснити підрахувавши набрані бали за виконані завдання. Відповідно до цього студенти діляться на дві підгрупи: ті, хто отримують оцінку "зараховано" (pass) і ті, у кого оцінка становить менше, ніж допустиме критичне значення (такі студенти отримують "не зараховано" (fail)). Останнім кредит не нараховується.

Система поділу першої підгрупи здійснюється наступним чином: після завершення вивчення дисципліни студентів диференціюють у порядку зниження успішності, зіставляючи одне з одним. Так здійснюється градація шкали ECTS grading scale.

Рівень, який отримує той, хто навчається, залежить від його місця в рейтингу. Оцінці «А» (10% від кількості, яка склала успішно) відповідає відмінний результат із допущенням незначних помилок. Оцінювання за шкалою «В» (25%) отримують при рівні знань вищому за середній і допущенні деяких неточностей. Результат «С» (30%) припускає ряд недоліків. Недопрацювання і суттєві помилки відповідають шкалі «D» (25%). «E» (10%) виставляють, якщо вимоги до відповіді студента мінімальні.

Оцінка "не зараховано" теж має рівні оцінювання: FX і F (потреба в додатковому опрацюванні матеріалу й значна додаткова робота відповідно).

Оскільки основою для ECTS grading scale є статичний підхід, необхідною кількістю для розподілу студентів у групі є 30. Перевагою при розрахунку є більша кількість осіб, що навчаються.

Дана система оцінювання (ECTS) має широке застосування у всьому світі. Вона характеризується значними перевагами, серед яких можна виділити наступні:

- можливість "прозоро" оцінити навчальні досягнення студентів;
- реальна оцінка навантаження при засвоєнні різноманітних видів навчальної діяльності;
- змога простежити сумісність навчальних планів у закладах методом їх структуризації.

Проте є певні труднощі (суб'єктивного й об'єктивного характеру) при переході вищих навчальних закладів на шкалу ECTS grading scale. Для статичного підходу, яким керується цей метод оцінювання, потрібне відповідальне ставлення студентів до навчання і спрямованість на максимальне досягнення результату, щоб у майбутньому отримати престижну роботу. Зважаючи на методіку побудови даної системи оцінок, вона не може базуватися на основі фіксованого розподілу.

Залік, як фінальна форма контролю знань, може бути зданий або ні. Виходячи з цього, ми не можемо простежити якість засвоєного навчального матеріалу, оцінюванню підлягає лише його кількість. Оскільки вітчизняні навчальні заклади мають невелику кількість академічних груп, статичний метод розподілу не є доцільним, адже для нього потрібна значна частина осіб, що навчаються.

1.2. Існуючі системи обліку успішності студентів

Важливо проаналізувати існуючі системи оцінювання й обліку студентів для розробки власної системи. Нижче буде розглянуто програмні продукти, котрі частково чи повністю задовільняють вимоги, описані вище, та застосовуються в

українських навчальних закладах, а також відповідають вимогам оцінювання системи ECTS.

В результаті огляду існуючих систем буде розглянуто, як вебсервіси із зовнішньою синхронізацією даних, так і програми, котрі працюють локально на пристрої викладача.

1.2.1. Застосунки для локального використання

Нижче буде розглянуто застосунки для локального використання викладачами (рис. 1.1), а саме андроїд-програми, котрі дозволяють вести облік успішності студентів.

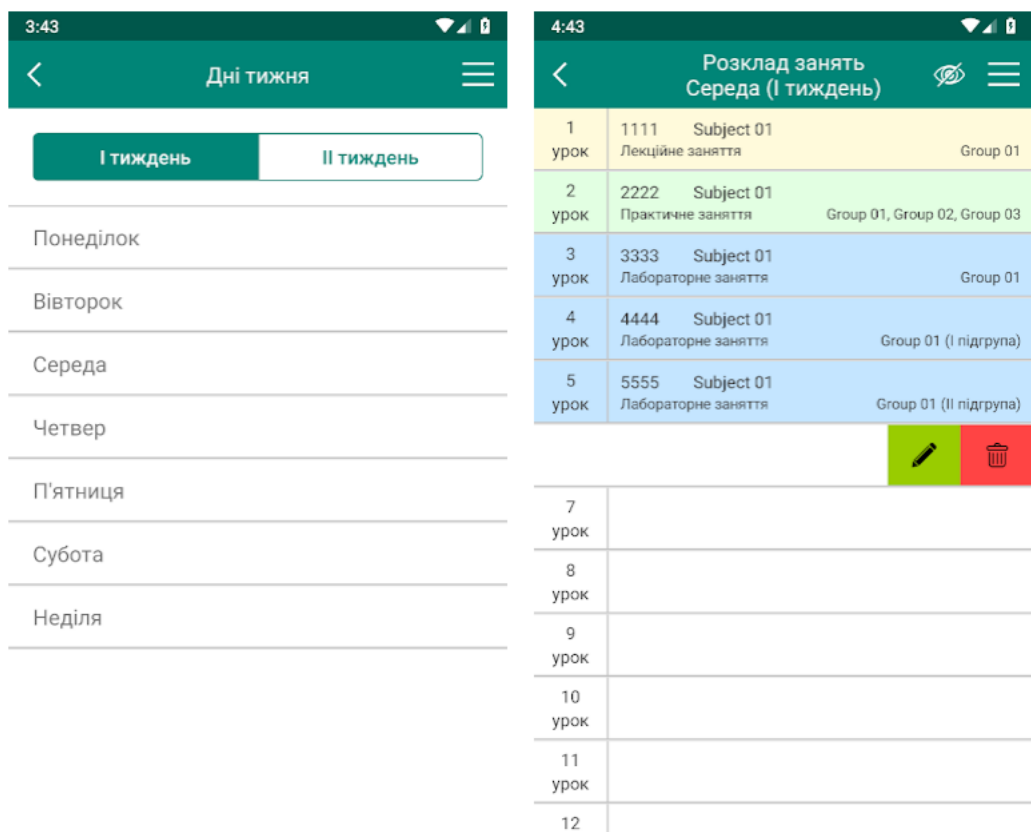


Рисунок 1.1 – «Журнал викладача»

Автор програми «Журнал викладача» зосереджує увагу на те, що створений ним додаток допомагає викладачеві контролювати присутність студентів на заняттях, а також простежити їхні успіхи із навчальної дисципліни.

Багатофункціональність проекту полягає у наступних характеристиках:

- розширена версія програми немає обмеження у кількості груп, підгруп чи осіб, які навчаються;
- з CSV-файлу можна здійснити перенесення списку груп, підгруп, студентів;
- створений варіант дозволяє експортування списку груп у формат Excel.

The screenshot shows the SJournal application interface. At the top, there's a navigation bar with icons for 'ГРУППА', 'ДИСЦИПЛИНА', 'ВИД ЗАНЯТИЯ', 'НОВОЕ ЗАНЯТИЕ', and 'РЕЙТИНГ'. Below this is a table with columns for 'Группа', 'Дисциплина', and 'Вид занятия'. The 'Дисциплина' column is expanded, showing a list of subjects: АПЭМА, МНЭМА, МП, НДЭМА, ОПЭ, and ТОЭ 3ч. The 'ТОЭ 3ч' subject is selected. Below this, there's a table with columns for '№', 'Ф.И.О.', and various dates/semesters. The table contains data for 9 students, with grades ranging from 4 to 6. The interface is in Russian.

№	Ф.И.О.	Л6 1 24.09.2013	Л6 2 01.10.2013	Захист Л6 1 08.10.2013	Захист Л6 2 15.10.2013	Л6 3 22.10.2013	Л6 4 29.10.2013	Л6 5.1 05.11.2013	Л6 5.2 12.11.2013	Л6 5 19.11.2013	Л6 6 26.11.2013
1	Белый Л.Г.			4	4						
2	Богданов С.С.			4	4	н	н	н			н
3	Буц В.В.	н	н	4	4	н	н	н	н	н	н
4	Волошин А.Ю.		н	4	4		н		н	н	н
5	Иванов И.И.		н	4	4				н	н	
6	Крамарев В.С.	н	н	4	4	н	н	н	н	н	н
7	Кулинич А.Ю.			4	4						
8	Петров Д.П.			4	4	6	н		н		
9	Потапов Ю.В.			4	4	н	н	н			н

Рисунок 1.2 – «Журнал преподавателя»

Програмний продукт “Журнал преподавателя” (рис. 1.2) розміщений на електронному джерелі Google Play Market. Додаток характеризується наявністю багатоманітних функцій, можливістю взаємодії комп’ютерних пристроїв та програм. Присутність засобів для заповнення довідників дозволяє формувати робочі записи. Засновник системи вказує, що вона створена для використання викладачами та є інструментом контролю успішності й відвідування занять учнями.

Додатковими можливостями, які її виділяють, є наступні:

журнал містить графу «середнє значення»;

налаштування зведених даних;

наявність графічного представлення журналу в формі таблиць;

той, хто навчається, може одразу ж входити в декілька груп чи підгруп;

можна здійснити імпортування даних про студентів із файлів;

база відомостей підлягає збереженню і відновленню;

записи в журналі роблять згідно з календарем, тобто має вказуватись група, скорочена назва дисципліни, різновид заняття (для прикладу: 3 пара, практична робота, вища математика (Вм), група ЕТ-02-1; скорочено цей запис подають так: 3п.Вм, ЕТ-02-1, пр.р.)

	Сентябрь 2013										← К текущей дате	
Ученик	2				Пн	2				Пн	3	Ср.
	ОТВ	ПОВ	ДЗ	К/Р	ОТВ	ПОВ	С/Р	К/Р	ДЗ	С/Р		
Бушин Юрий	2		5	2 2				2	4+	5	3.62	
Васнецов Аристарх	5	Н/А	5-	5 5	5	5	5				5.00	
Иванов Дмитрий	6		5	3 4		4-		2		2	3.87	
Иванова Марина	5		5	5		4		2		4	4.17	
Ильгова Ольга	7		5	5 5				2		5	4.00	
Кузнецов Артемий	5	ОСВ	5	5 3		5				1	3.80	
Лихачев Игорь				5 5							5.00	
Поспелов Глеб	4		5	5 5				2		5	3.57	
Радужкина Анна	6		5	5 5				2		7	4.07	
Урбина Светикова	4		1-	5 4				2		4	3.44	

Рисунок 1.3 – «Мобільний журнал»

Характеризуючи додаток «Мобільний журнал», засновники вказують на його перевагу, адже викладачі можуть здійснити записи в режимі дійсного часу.

За допомогою цього програмного продукту (рис. 1.3) можна робити нотатки про роботу студента на заняттях, створювати й давати домашні завдання, оцінювати студента, відмічати присутність, здійснювати запис теми навчального заняття, а також можливість його редагування, створювати роботу, вносити корективи чи видаляти її.

Інструкція, подана в журналі, ознайомлює користувача із засобами, методами й правилами взаємодії між елементами. Одразу ж після відкриття викладач зможе простежити особливості користування програмою. Вона містить додаток «Реалізований підрахунок середнього балу студентів», що значно полегшує роботу.

Особливістю створеної системи є також зручний перехід від одного до іншого режиму виставлення оцінок.

Для того, щоб почати роботу над додатком, варто звернути увагу на деталі у використанні:

- програму потрібно завантажити на мобільний пристрій;
- планшетний прилад, з якого здійснюють користування журналом, повинен мати діагональ екрана 7 і операційну систему Android v. 4.0. та вище;
- працювати із додатком потрібно тільки при підключеному до мережі Інтернет пристрої.

1.2.2. Вебсервіси для обліку успішності студентів

Програма «Електронний кампус НТУУ КПІ» (рис. 1.4) являє собою платформу, яка допомагає здійснювати контроль за успішністю тих, хто навчається у НТУУ «КПІ ім. Ігоря Сікорського». Її створення має на меті допомагати учасникам навчального процесу (студентам та іншим робітникам закладу).

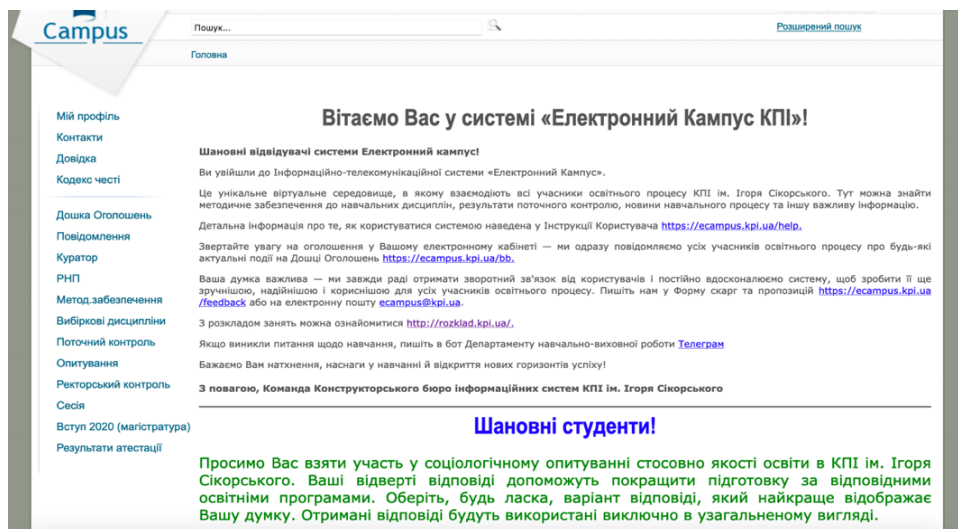


Рисунок 1.4 – Електронний кампус НТУУ КПІ

Розроблений ресурс задовільняє такі потреби користувачів:

- інформування щодо подій чи деталей навчального процесу;
- можливість зворотно отримати відповідь (зв'язок викладач-студент);
- використання цієї системи створює одне джерело інформації для всіх членів університету.

Як засвідчили опитування серед студентів, програмним забезпеченням користуються тільки 14%, що є дуже малим показником. Причиною цього може

бути ігнорування думок тих людей, які мають вплив на кінцевий добір користувачів.

Серед недоліків системи потрібно відзначити застарілий інтерфейс. Особливості оформлення будь-якого інформаційного джерела одразу привертають увагу користувача, адже це перше, з чим він ознайомлюється при вході. Тим більше, якщо програма створена для задоволення потреб соціуму.

Оскільки інформація, що подається у програмному продукті, може бути об'ємною, вона має бути представлена в зручному вигляді для користувача. Тим паче, якщо розробник ресурсу має на меті його довготривале використання.

Не варто знецінювати значення інтерфейсу, адже правильна й чітка розробка може забезпечити постійність у користуванні джерелом і дозволить уникнути відсіювання осіб, які вже ним користуються.

Серед хиб, що містить даний ресурс, треба виділити й відсутність оновлення інформації.

Проте саме актуальні відомості протягом всього процесу навчання дадуть змогу підготувати справжніх фахівців. Через вище вказані особливості в побудові програми можна простежити відмінності між її описом і тим, що насправді знаходиться на ресурсі.

Відсутність оновлених даних чинить вплив на якість навчального процесу, тому на цей недолік потрібно звернути особливу увагу.

Для обчислення успішності студента із вивченої дисципліни необхідна робота з ресурсами, що стосуються сфери навчального курсу. Інформація повинна бути оформлена чітко й зрозуміло, як для студента, так і для викладача. Це забезпечить результативність у процесі вивчення. Перевагою було б використання засобів, які зроблять можливим збереження відомостей про поточну й підсумкову успішність.

Для цього доцільним буде клас програмних засобів LMS. Він має ряд ресурсів, необхідних для оцінювання, а також контролю за процесом передавання і засвоєння знань. Використання LMS значно спрощує роботу для користувача, полегшує можливість контролювання, дозволяє досягти

ефективних результатів при вивченні дисципліни, надає тому, хто ним користується, сукупність необхідних можливостей.

Подібні системи особливо популярні при дистанційному навчанні, хоча їх застосування знайшло себе і в освіті вищої школи. Для результативного й повноцінного користування подібними джерелами потрібні значні затрати ресурсів, щоб освоїти вже наявні функціональні особливості, а також присутність елементів, що наповняють систему відповідною інформацією.

Прикладом, що відповідає таким вимогам, є Moodle-платформа, яка допомагає управляти навчальним процесом. Вона створена для використання у безкоштовному секторі програмних засобів.

Необхідно зрозуміти, що подібні системи для обліку успішності студентів мають певні недоліки. В результаті чого я розглянув можливі варіанти для вирішення труднощів, які виникають при користуванні. Слід зрозуміти – такі помилки при створенні програмних систем не дають змоги їм стати невід’ємним елементом навчального процесу.

1.3. Задачі, котрі мають бути реалізовані у вебсервісі для обліку успішності студентів

Вище ми дослідили систему оцінювання, яка має бути використана в системі обліку успішності, тож потрібно сформулювати вимоги до програмного продукту, котрий працюватиме в описаних умовах. Також ми дослідили переваги й недоліки існуючих сервісів і можемо визначити завдання, які необхідно вирішити, щоб уникнути проблем уже наявних сервісів та зробити корисний продукт для користувача. Для досягнення поставленої мети в роботі визначені наступні завдання:

- на основі аналізу сучасних розгалужених систем обліку та контролю різних типів побудувати архітектуру системи, оптимізовану для використання у вузах;
- інтегруватися з онлайн табличним сервісом;

- розробити універсальний спосіб аутентифікації різних типів користувачів використовуючи поточні дані університетської системи обліку;
- розробити програмну реалізацію по видачі списків студентів і електронного журналу в онлайн-таблицях для викладача, реалізувати ендпоїнти для отримання актуальних даних та вебхук або полінг для прямого сповіщення студентів, або програмного рішення для сповіщень.

Висновки до розділу

Розглянуто систему оцінювання, під стандарти якої буде розроблятися програмний продукт, а також існуючі реалізації сервісів для ведення обліку та інформування студентів про успішність.

Визначено переваги та недоліки існуючих систем, котрі є підґрунтям для розробки вебсервісу для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.

2. ВИБІР ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБСЕРВІСУ ДЛЯ ОБЛІКУ УСПІШНОСТІ

Для розробки вебзастосунку потрібно обрати стек технологій, що складається із таких елементів:

- мови програмування для бекенд логіки вебсервісу;
- фреймворку для бекенд розробки із вбудованим засобом взаємодії з БД – ORM;
- бази даних;
- бібліотеки для інтеграції з API таблиць;
- месенджеру та бібліотеки до його API для сповіщень студентів засобом чат-бота.

Почнемо з вибору мови програмування.

2.1. Мова програмування Python

Python – мова програмування, яка характеризується універсальністю у використанні та якісним рівнем організації. Її прерогативою є чітко зрозумілий шифр, а також продуктивність програмних завдань. Мова Пітона побудова таким чином, щоб користувач міг освоїти її за короткий проміжок часу. Основа Python має зрозумілу побудову й широкий спектр вбудованих бібліотек. Відповідно дані ознаки дають можливість застосовувати значний набір корисних можливостей і функцій. Досліджувана мова програмування є інструментом, що допомагає при створенні WEB-сервісів та має широке застосування у прикладному програмуванні.

Пітон може підтримувати різноманітний вибір стилів потрібних для створення додатків. Варто зазначити, що ця мова доречна при об'єктно-орієнтованому й функціональному програмуванні (рис. 1.5).

Еталоном для тлумачення Python є CPython, що написаний на С. Його виділяє вільне та відкрите програмне забезпечення, а також безкоштовний доступ. Цей інтерпретатор підтримує безліч відомих усім платформ.

Він весь час зазнає динамічних змін, оновлень, які появляються раз у два роки. СPython немає певних стандартів кодування. Це стосується ANSI та ISO, адже їх робота забезпечується за його допомогою. Особливості в побудові Python дають можливість застосувати цю мову буквально на всі операційні системи, використовуючи при цьому як великі сервери, так і кишенькові комп'ютери.

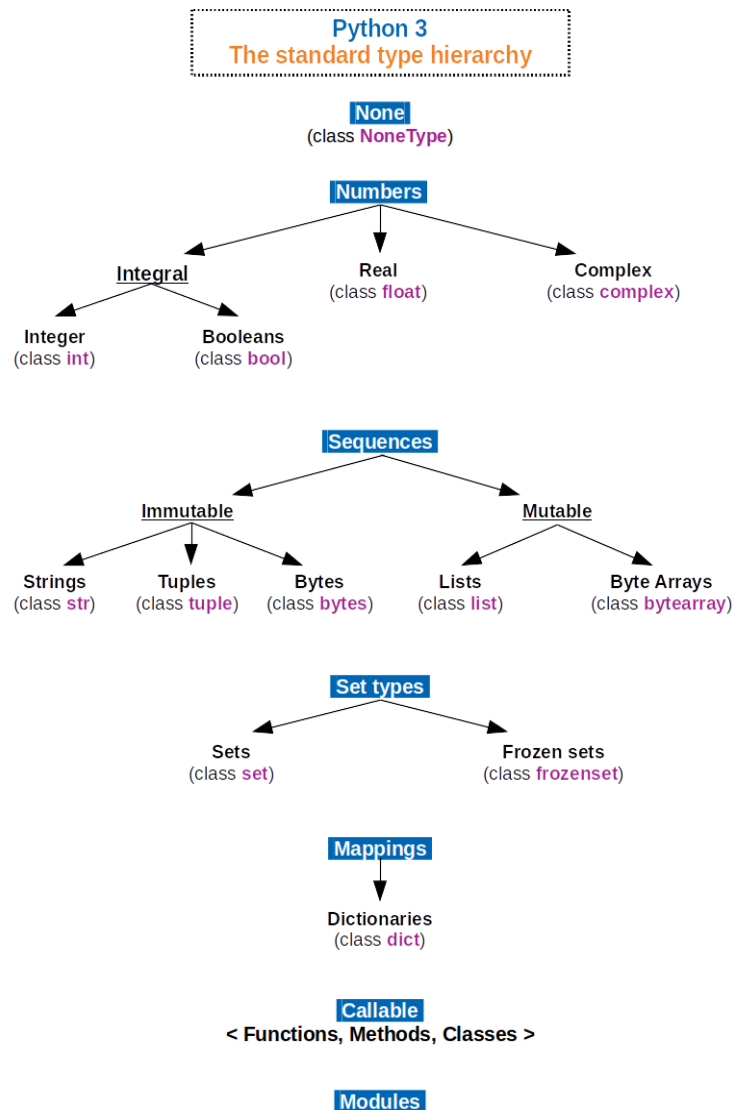


Рисунок 2.1 – Типи даних Python

Застарілі платформи вилучаються і більше не підтримуються ядром цієї мови програмування. Тобто версії Пітон, починаючи з 2.6, не підтримують Windows 95, 98, ME. Для роботи додатку із застарілими операційними системами є змога скористатись старішими версіями, проте у них відсутні сучасні інструменти. Ці версії час від часу отримують інформацію, що автоматично вносить певні зміни (так звані патчі або латки). Досліджувана мова підтримує виконання завдань із віртуальною машиною Java.

2.2. Фреймворк для веброзробки на мові Python. Django

При виборі вебфреймворку важливо правильно дотримуватися балансу в простоті розробки, швидкості виконання програми та готових рішеннях. Було досліджено Flask, Aiohttp, Django.

Серед перших двох за швидкістю виконання та оновлюваністю технології перемагає Aiohttp, але ця технологія не містить готової ORM всередині з вебпрезентацією на адмін-панель із готовими рішеннями для адміністрування ролей, як Django.

Django (рис. 1.6) – це веб платформа високого рівня, яка дає можливість створити сайти безпечні у користуванні. Розробники відповідально поставились до створення програми й доклали максимум зусиль. Саме тому ця платформа дозволяє зосередитись більше на написанні самого веб-додатку, аніж на вигаданні деталей веб-розробки

Безперечно, перевагами Django є наявність, як безкоштовної, так і платної служби підтримки й відкритий для користувача вихідний пароль. Сам програмний продукт є безплатним. Неможливо не відмітити й відмінну документацію та активних учасників платформи

Програмне забезпечення, створене за допомогою цього додатку є повним. Django користується принципом "батарейки в комплекті", адже може задовольнити всі потреби розробника. Така злагоджена система програми дозволяє всім елементам здійснювати одну і ту ж роботу одночасно, відповідає вимогам проектування, володіє вичерпною та актуальною інформацією.

Платформу з-посеред інших виділяє різносторонність. Django може допомогти у створенні соціальних мереж, сайтів новин, систем управління контентом і wiki. Ця особливість дозволяє продукувати практично всі види вебсайтів. Багатофункціональність платформи підкреслює й те, що вона підтримує контент різноманітних форматів: HTML, RSS-канали, XML, JSON. Таким чином Django відповідає потребам багатьох клієнтів й надає вирішення будь-якого запиту. При необхідності можна розширити цей програмний продукт всередині за допомогою сторонніх елементів.

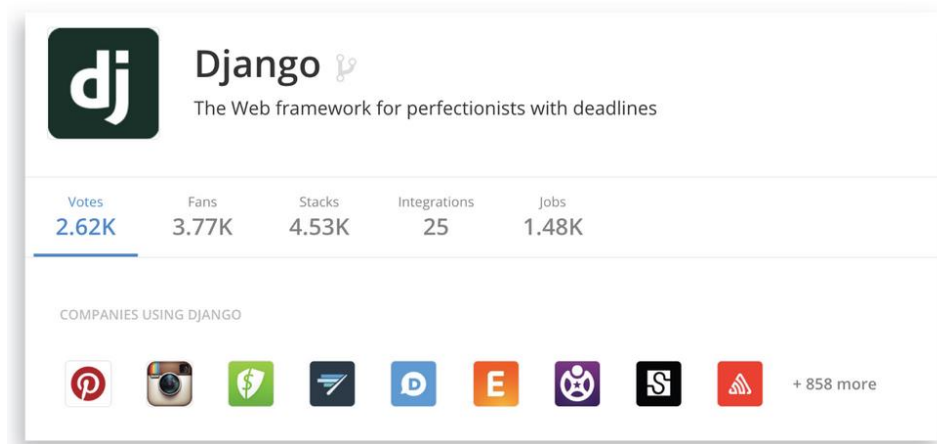


Рисунок 2.1 – Компанії, які використовують Django

Програма містить фреймворк, що працює за принципом «робити правильні речі». Це автоматично захищає сайт і характеризує Django як безпечну для роботи платформу. Така система побудови допомагає уникнути проблем із безпекою. Облікові записи й паролі клієнтів знаходяться під захистом, а самі дані не зберігаються у файлі cookie. Тобто cookie має у своєму доступі тільки ключ, натомість фактичні відомості зберігаються у базі даних. Це стосується безпосереднього збереження паролів замість хеша пароля.

Сутність хеш-паролю полягає у його обробці через криптографічну хеш-функцію. Після введення паролю Django може перевірити його правильність. Для цього використовується хеш-функція, яка полягає у порівнянні вихідного результату із вже збереженим значенням хеша. Якщо ж зломиснику і вдасться отримати доступ до хеш-значення, то завдяки особливостям побудови Django, він не зможе легко визначити пароль.

Розробники платформи створили захист від різноманітних подразників, таких як: міжсайтовий скриптинг, SQL-ін'єкція, підробка запитів, клікджекінг. Django має особливу побудову, бо кожна його частина сприймається як ціла, незалежна від інших, тому це дає можливість легко замінити сегменти чи робити певні поправки. Така структура має назву "shared-nothing".

Оскільки усі компоненти системи є самостійними, то Django може масштабуватись при збільшенні трафіку. Це досягається додаванням інструментів різного рівня: серверів додатків, баз даних, серверів кешування. Для прикладу, всім відомі соціальні мережі Instagram і Disqus успішно збільшили

Django. Ця платформа є зручною у використанні, адже її код створений із врахуванням вимог та шаблонів проектування, що заохочує користувача у свою чергу створити пароль, який він і надалі буде використовувати. Система коду побудована за принципом "Don't Repeat Yourself", тобто закликає не повторюватись. Тут відсутній повтор, що значно скорочує об'єм пароллю. Для полегшення роботи користувача Django групує пов'язані між собою функціональні можливості, створюючи додатки, які можна повторно використати. Паролі також підлягають систематизації, тому Django створює модулі (відповідно до шаблону Model View Controller (MVC), (рис. 1.7).

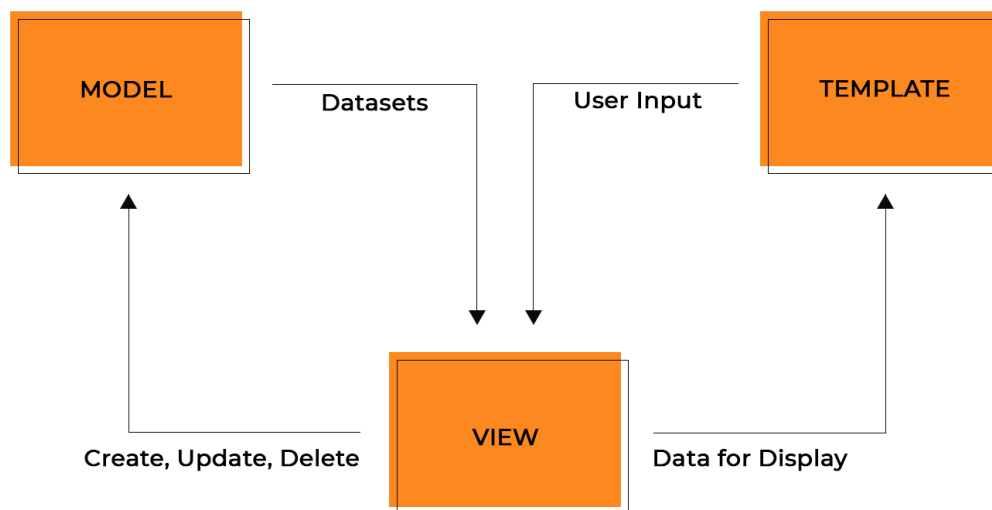


Рисунок 2.3 – Глобальна Архітектура Django

Досліджувана платформа є переносною. Django створений на Python, а, як відомо, Пітон працює на багатьох платформах, відповідно користувач не прив'язаний до якоїсь певної платформи, тому може здійснити вхід і з таких версій, як Linux, Windows, Mac OS X. Універсальність цієї системи пояснюється підтримкою безлічі веб-хостингів, вони ж у свою чергу надають певну документацію і сукупність систем для розміщення сайтів Django.

Для одержання запиту при користуванні традиційним інтернет-сайтом потрібно отримати HTTP респонз. Веб-переглядач чи інший клієнт передає отримані дані вебзастосунку. Розробка необхідних завдань відбувається на основі URL-адреси або ж інформації у POST чи GET запитах одразу після отримання запиту. Уніфікований локатор ресурсів може виконувати різноманітні завдання, читати й зберігати відомості з бази даних. Між

застосунком та веб-переглядачем відбувається взаємодія. Вона полягає у поверненні відповіді браузеру шляхом створення сторінки HTML для подальшого відображення веббраузера. Отримана інформація поміщається у HTML шаблон (рис. 2.4).

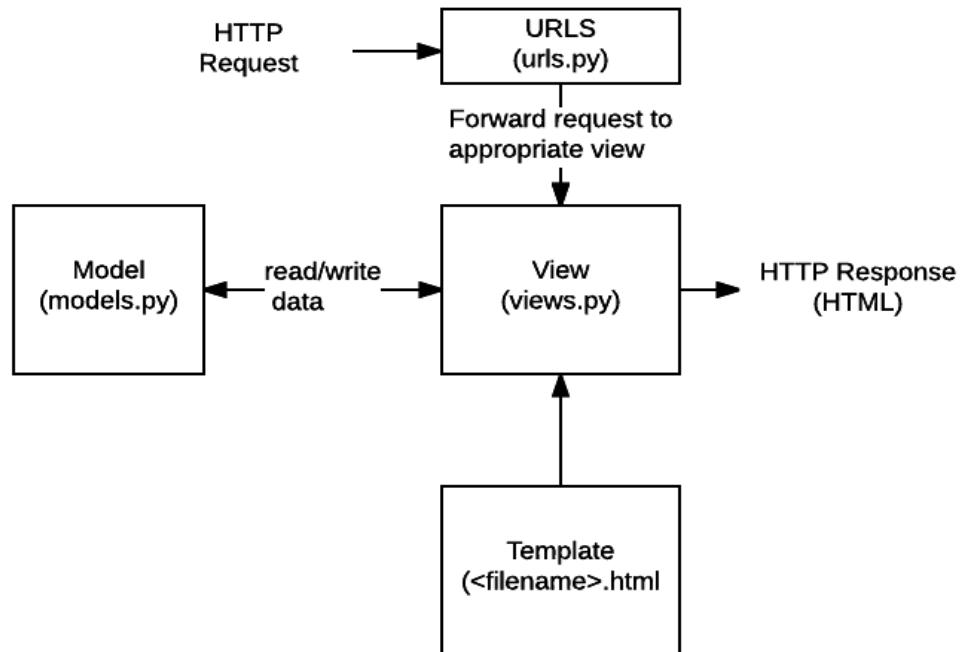


Рисунок 2.4 – Архітектура застосунків Django

Застосунки, які створюють, використовуючи платформу Django, групують між собою код. Він натомість опрацьовує усі дії й утворює з них окремі файли.

URLs:

Запити з кожної URL-адреси виконуються за допомогою однієї функції. Але все ж значно зручніше писати окремі функції для обробки кожного ресурсу. Усі HTTP-запити переадресовуються у відповідне подання на основі URL-адреси. Інструментом для перенаправлення є URL-mapper. З його допомогою можна витягувати дані з адреси ресурсів. Відповідно до шаблону URL-mapper передає інформацію у вигляді аргументів у відповідну функцію.

View:

Однією із функцій, що здатна опрацьовувати запити, є View. У своїй компетенції вона має доступ до даних через моделі, необхідних для передачі відповіді у шаблони. Основна функція View полягає в отриманні HTTP-запитів і поверненні відповідей.

Models:

За допомогою моделей можна визначити будову даних застосунку. Models є об'єктами Python і мають систему елементів, потрібну для додавання, зміни, видалення запитів, а також їх виконання.

Templates:

Для визначення структури й розмітки сторінки (для прикладу HTML-сторінки) використовують Template (шаблон), іншими словами текстовий файл. За допомогою HTML-шаблонів View може активно створювати HTML-сторінки й заповнювати їх даними. Шаблон здатний визначати структуру різних файлів, адже він підходить для роботи із різнотипними блоками інформації.

Файл `urls.py` є місцем для співставлення URL-адрес. Нижче можна простежити, як `urlpatterns` (диспетчер для зіставлення) з'ясовує відповідність між URL-шаблонами й відповідними функціями відображення. Після чого виникає й передається в запит асоційована функція відображення (`views.index`). Це відбувається у випадку, коли HTTP запит, який отримали, відповідає шаблону (як `r'^$'` на зображенні).

```
urlpatterns = [ url(r'^$', views.index), url(r'^([0-9]+)/$', views.best), ]
```

Вебзастосунок після прийняття запиту від клієнта здійснює повернення інформації у вигляді HTTP-відповідей. Так виникає `views`, що є основою будь-якого застосунку.

У прикладі нижче зображена мінімальна функція відображення `index()`. Її міг викликати перетворювач URL. Це відбувається наступним чином: `index` отримує об'єкт `HttpRequest` як параметр `request`, а натомість повертає `HttpResponse`. Виходить, користувач нічого не робить із запитом, адже відповідь надсилається незмінно запрограмованим пошуковим рядком.

```
## filename : views.py ( Django view functions )
from django.http import HttpResponse
def index( request ):
    # Отримати HttpRequest - параметр запиту
    # Виконати операції, використовуючи інформацію із запиту.
    # Повернути HttpResponse
    return HttpResponse ( ' Hello from Django ! ' )
```

Усі відображення View зберігаються у `views.py`.

2.3. База даних PostgreSQL

PostgreSQL – об'єктно-реляційна система управління базами даних (СУБД). Вона підходить для програм із вільним програмним забезпеченням (MySQL, Firebird, SQLite) та програм комерційного типу (Oracle Database, Microsoft SQL Server, IBM DB2).

Автором серверу є Майкл Стоунбрейкер, підготовка й запуск системи належить групі розробників. Постгрес підтримує такі ОС, як Unix, Linux, Windows й написана мовою С. Ця система управління має ліцензію (PostgreSQL license) та власний сайт.

Сервер не знаходиться під контролем однієї компанії, адже в його розробці можуть брати участь безліч користувачів чи компаній. Це відрізняє PostgreSQL від інших подібних проектів із вільним програмним забезпеченням.

Користувач, якому потрібно скористатись цією системою, отримує її у вигляді набору текстових файлів.

Для запуску постгрес треба здійснити процес перетворення комп'ютерної програми у відповідний код на ПК, після чого скопіювати отримані файли в папку.

До основних можливостей PostgreSQL належать функції. Вони дають дозвіл на виконання деякого коду сервером даних. Для написання функцій використовують SQL. Ця мова програмування складається з примітивних операторів (галуження та цикли (рис. 1.9)). Якщо для написання функції використовують мову програмування, яку підтримує постгрес, то тоді функція буде гнучкішою. PostgreSQL працює із мовами розробки сценаріїв (PL/Perl, PL/Python, PL/Tcl, PL/Ruby, PL/sh), класичними мовами програмування (С, С++, Java), вбудованою мовою PL/pgSQL.

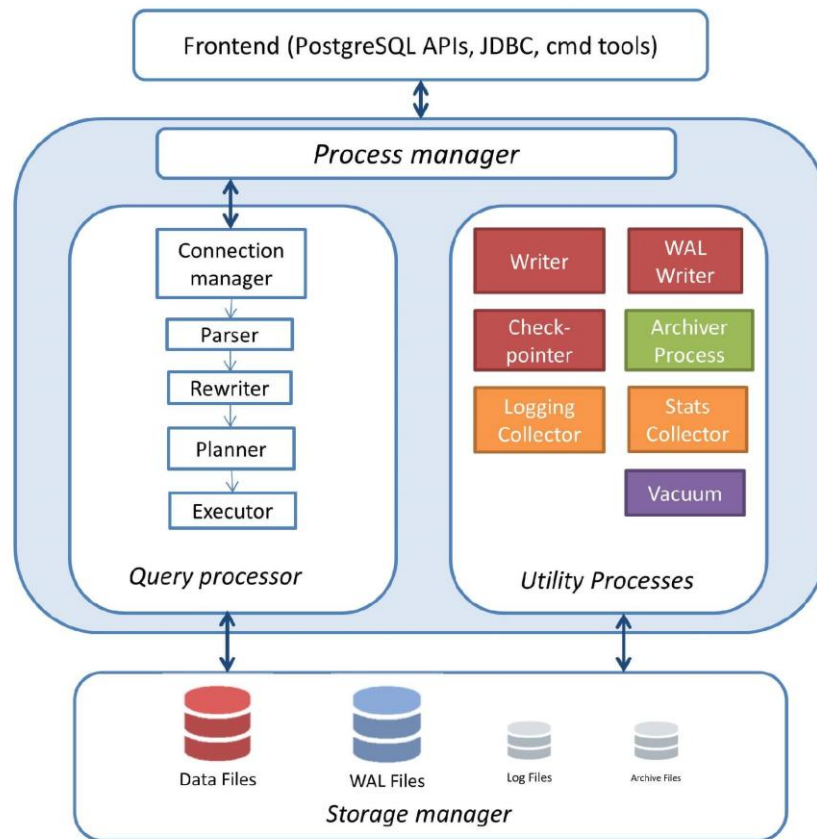


Рисунок 2.5 – Архітектура PostgreSQL

Ця система керування підтримує різноманітні індекси, такі як GiST, GIN, хеш, R-дерево, B-дерево. Можливе створення нових типів індексів користувачем. База даних може бути видозмінена одночасно кількома користувачами, це вказує на багатоверсійність PostgreSQL. Модифікація задовольняє потреби ACID, після чого блокувати зчитування вже немає потреби.

PostgreSQL підтримує мережеві, числові, двійкові, символьні, OID та псевдотипи, а також типи "дата/час" і грошовий тип. Цілі типи з фіксованою і нефіксованою крапкою, грошовий, булевий тип, перерахування, геометричні примітиви, масиви, JSON-, XML-дані, UUID-ідентифікатор теж підтримуються цією системою.

Характеристики та набори полів передаються від однієї до іншої таблиці. Одна з них є батьківською, інша – породженою. При здійсненні запитів до батьківської, дані від похідної таблиці машинально братимуть участь у цьому процесі. З цього випливає, що PostgreSQL властиве успадкування.

Тригери – це функції, що запускаються DML-операціями. Для прикладу можна навести операцію INSERT: тригер запускається, якщо треба перевірити запис на відповідність деяким умовам. Для писання тригерів підходять різні мови програмування. Їх характеризує зв'язок із визначеною таблицею. Якщо тригери численні, то їх виконання здійснюється в алфавітному порядку.

2.4. Python обгортка навколо API Google Sheets

Gsheets – це невелика обгортка навколо API Google Sheets (v4), щоб забезпечити зручніший доступ до Google Sheets зі сценаріїв Python. Сюди входить API, ідентифікатор клієнта OAuth як файл JSON і здатна створювати об'єкт таблиці. Використовує доступ до індексу (`__getitem__`), щоб отримати об'єкти Spreadsheet за їх ідентифікатором, або використовує метод `.get()` із URL-адресою аркуша. Дозволяє переглядати об'єкти аркушів для всіх електронних таблиць або отримувати електронні таблиці за заголовком за допомогою методів `.find()` та `.findall()`.

Об'єкти Spreadsheet – це колекції Worksheets, які забезпечують доступ до значень комірок за допомогою координат/зрізів електронних таблиць (наприклад, `ws['A1']`) та нульового положення комірки (наприклад, `ws.at(0, 1)`). Дає змогу зберігати Worksheets (або всі сторінки Spreadsheet) як файли CSV за допомогою методу `.to_csv()`. Створює `pandas.DataFrames` з робочого аркуша методом `.to_frame()`.

2.5. Месенджер для інтеграції сповіщень засобом чат-бота – Telegram

Платформою для чат-бота для сповіщення студентів було обрано Telegram. Telegram – клауд-месенджер, програмне забезпечення для смартфонів, планшетів та ПК, яке дозволяє обмінюватися текстовими повідомленнями, графічними та відеофайлами, а також безкоштовно телефонувати іншим користувачам програми.

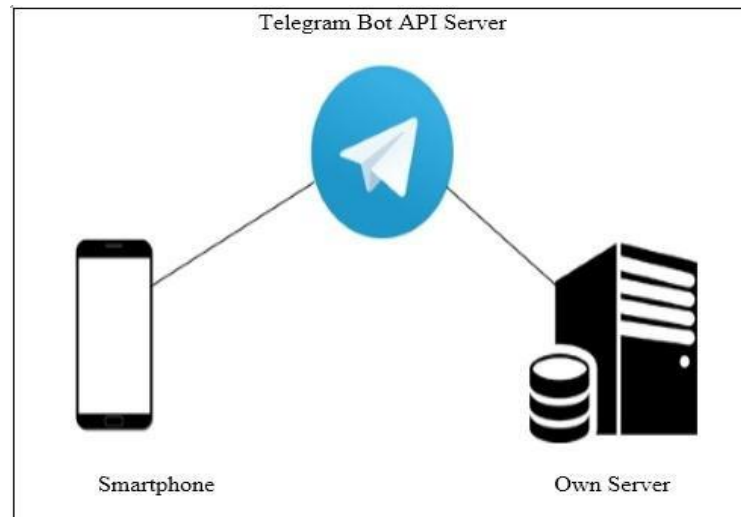


Рисунок 2.6 – Спрощена схема роботи телеграм-ботів

У Telegram працює платформа чат-ботів (рис. 2.6). Боти можуть виконувати різноманітні завдання, такі як пошук в інтернеті чи держреєстрах, покупки, платежі, розваги, модерація груп тощо.

У спілкуванні беруть участь користувач Telegram та комп'ютерна програма від стороннього розробника. Користувач може взаємодіяти з ботом за допомогою елементів інтерфейсу месенджера: надсилання повідомлень, натискання на команди та кнопки, використання онлайн-режиму. Telegram надає три способи взаємодії користувача з ботом: приватний чат (класичний спосіб), група й так званий онлайн-режим.

Висновки до розділу

Було обрано такі технології: мова програмування для вебу – Python. Мова програмування високого рівня, яка характеризується універсальністю у використанні та якісним рівнем організації. Для вебскладової застосунку – Django. Високорівневий вебфреймворк з ORM та веб-адмін з панеллю всередині. Платформою для чат-бота для сповіщення студентів було обрано Telegram. Верстка додаткових компонентів вебу буде реалізована на відкритій бібліотеці bootstrap. Інтеграція з гугл таблицями за допомогою API. Gsheets – це невелика обгортка навколо API Google Sheets (v4), щоб забезпечити зручніший доступ до Google Sheets зі сценаріїв Python.

3. РОЗРОБКА ВЕБСЕРВІСУ ДЛЯ ОБЛІКУ УСПІШНОСТІ

3.1. Розробка архітектури вебзастосунку

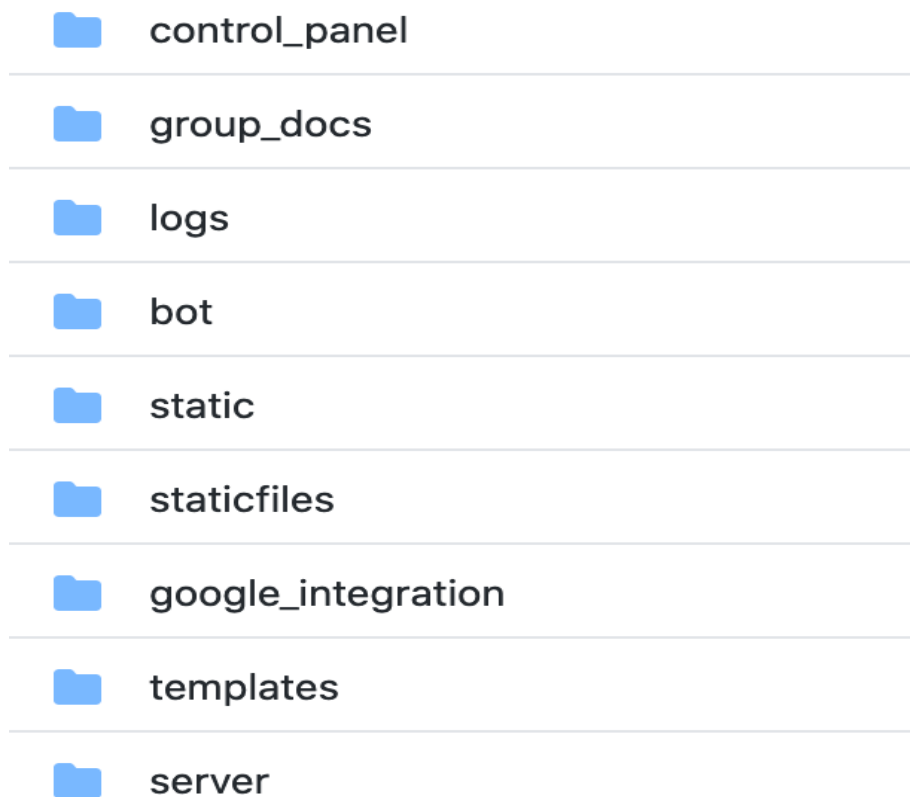


Рисунок 3.1 – Підпрограми.

Панель управління – `control_panel` розширення джанго адмін панелі для роботи з користувачами, підгрузки списків груп з ексель документів. Валідації даних при підвантаженні (рис. 3.1).

Підпрограма групових документів `group_docs` – використовується для відображення в'ю для підвантаження списків.

Налаштування логування `logs` – файл настройки логів.

Інтеграція з телеграм API, логіка сповіщень – `bot`.

Папка із статичними файлами каскадних таблиць стилів, джаваскрипту, бібліотеки для адаптивної верстки – бутстоап. `Static`.

Інтеграція гугл таблиць, отримання даних про оцінки, апдейт даних успішності – `google_integration`.

Папка основних налаштувань застосунку – `server`.

Групи та користувачі – стандартні моделі джанго, в яких вже закладено функціонал для розмежування ролей та обліку всіх користувачів проекту. Більш

детально розберемо ключові таблиці. Інтеграція з телеграм API, логіка сповіщень – bot.

Папка із статичними файлами каскадних таблиць стилів, джаваскрипту, бібліотеки для адаптивної верстки – бутстоап. Static.

Нижче бачимо опис таблиці викладачів.

```
class Teacher(models.Model):
    link = 'Редагувати'
    class Meta:
        db_table = 'Teachers'
        verbose_name = 'Викладач'
        verbose_name_plural = 'Викладачі'
        teacher_full_name = models.CharField(max_length=200, default=None,
        verbose_name="ПІБ")
        teacher_department = models.ForeignKey('control_panel.Department',
        on_delete=models.CASCADE, related_name='+', blank=True, null=True,
        verbose_name="Кафедра")
    def __str__(self):
        return self.teacher_full_name
```

Нижче бачимо опис таблиці кафедр.

```
class Department(models.Model):
    link = 'Редагувати'
    class Meta:
        db_table = 'Departments'
        verbose_name = 'Кафедра'
        department_name = models.CharField(max_length=100, default=None,
        verbose_name="Назва")
        department_head = models.ForeignKey(Teacher, on_delete=models.CASCADE,
        related_name='+', blank=True, null=True, verbose_name="Зав. Каф.")
    def __str__(self):
        return self.department_name
```

Нижче бачимо опис таблиці студентів. Як бачимо, кожен студент зав'язаний на користувача аплікейшна.

```
class Student(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, blank=True)
    link = 'Редагувати'
    class Meta:
        db_table = 'Students'
        verbose_name = 'Студент'
        student_active = models.BooleanField(default=True)
    objects = StudentManager()
```

```

student_full_name = models.CharField(
    max_length=200,
    default=None,
    verbose_name="ПІБ",
    blank=True,
    null=True
)
student_group = models.ForeignKey(
    'control_panel.Group',
    on_delete=models.SET_NULL,
    verbose_name='Група',
    blank=True,
    null=True
)
student_notes = models.TextField(
    default=None,
    blank=True,
    verbose_name="Примітки",
    null=True
)
def __str__(self):
    return self.student_full_name
def get_listed_full_name(self):
    return self.student_full_name.split(' ')

```

Нижче бачимо опис таблиці предметів.

```

class Discipline(models.Model):
    link = 'Редагувати'
    class Meta:
        db_table = 'Disciplines'
        verbose_name = 'Предмет'
        verbose_name_plural = 'Предмети'
    discipline_name = models.CharField(max_length=100, verbose_name="Назва")
    discipline_department_name = models.ForeignKey(Department,
on_delete=models.CASCADE, related_name='+', blank=True, null=True,
verbose_name="Кафедра")
    def __str__(self):
        return self.discipline_name

```

Нижче бачимо опис таблиці груп.

```

class Group(models.Model):
    link = 'Редагувати'
    class Meta:
        db_table = 'Groups'

```

```

        verbose_name = 'Група'
        group_id = models.IntegerField(default=None, verbose_name="Номер групи")
        group_department = models.ForeignKey(Department, on_delete=models.CASCADE,
        related_name='+', blank=True, null=True, verbose_name="Кафедра")
        def __str__(self):
            return str(self.group_id)
        def get_students(self):
            return Student.objects.filter(student_troop=self)

```

3.2. Інтеграція табличного сервісу. Локальне завантаження списків груп

Для завантаження списків груп, розроблено модуль StudentUploader, котрий дозволяє зі стандартизованої Excel таблиці вивантажити студентів груп в базу даних, створити користувача для кожного студента, а також згенерувати паролі для доступу до сервісу кожному користувачеві й записати в таблицю студентів, доповнивши поля.

```

@staticmethod
def password_generator(size=6):
    password = ''
    for i in range(size):
        password += random.choice(string.ascii_letters)
    return password

```

Вище бачимо складову завантажувача списків – метод генерації паролів.

3.3. Розробка інтерфейсів вебсервісу

Адміністративна панель моделей датибази та доступів користувачів генерується фреймворком Django автоматично, але потрібно в файлі `admin.py` описати, які поля відображати при фронт-енд репрезентації полів таблиць (рис. 3.2).

Адміністрування сайта

АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ		
Групи	+ Додати	Змінити
Користувачі	+ Додати	Змінити
КЛЮЧОВІ ТАБЛИЦІ		
Викладачі	+ Додати	Змінити
Таблиці оцінок	+ Додати	Змінити
Оцінки	+ Додати	Змінити
Предмети	+ Додати	Змінити
Студенти	+ Додати	Змінити

Панель реєстрації груп

[На панель](#)

Рисунок 3.2 – Адмін-панель вебсервісу.

```
from django.contrib import admin
from django.contrib.auth import get_user_model
from django.contrib.admin.actions import delete_selected as delete_selected_
from django.core.exceptions import PermissionDenied
from control_panel.models import *
from control_panel.forms import *
admin.site.site_title = 'Диплом'
@admin.register(Group)
class TroopLayout(admin.ModelAdmin):
    list_display_links = ('link',)
    list_display = (
        'group_id',
        'group_head',
        'group_department',
```



```

        'link'
    )
@admin.register(Teacher)
class TeacherLayout(admin.ModelAdmin):
    list_display = (
        'teacher_full_name',
        'teacher_department',
        'link'
    )
    list_display_links = ('link',)
@admin.register(Student)
class StudentLayout(admin.ModelAdmin):
    list_display = (
        'student_full_name',
        'student_group',
        'student_faculty',
        'student_grade',
        'show_marks',
        'student_state',
        'student_notes',
        'link'
    )
    search_fields = (
        'student_full_name'
    )
    list_filter = (
        'student_group',
        'student_faculty',
        'student_grade'
    )
    list_display_links = ('link',)

```

Також було розроблено кастомні інтерфейси для зручного управління вебсервісом викладачами (рис 3.3). Надалі будемо називати їх «Панель». На панелі викладача було розроблено наступні інтерфейси:

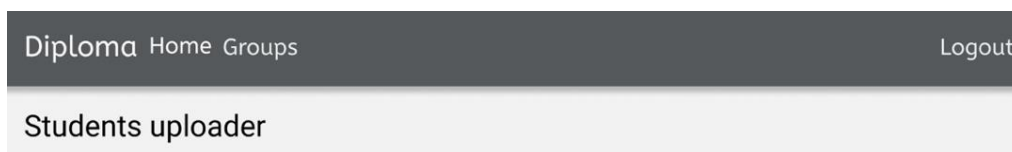


Рисунок 3.3 – Навбар сервісу.

Опис стилів навбару:

```
/* Nav */
```

```
position: absolute;
width: 730px;
height: 54px;

filter: drop-shadow(0px 4px 4px rgba(0, 0, 0, 0.25));
```

Нижче – метод підвантаження файлу із списком в адмінку.

```
@classmethod
def handle_uploaded_file(cls, file, file_name: str):
    with open(cls.PATH_FOR_SAVING + slugify(file_name, 'uk') +
'.xlsx', 'wb+') as destination:
        for chunk in file.chunks():
            destination.write(chunk)
        file_path = cls.PATH_FOR_SAVING + file_name + '.xlsx'
    return file_path
```

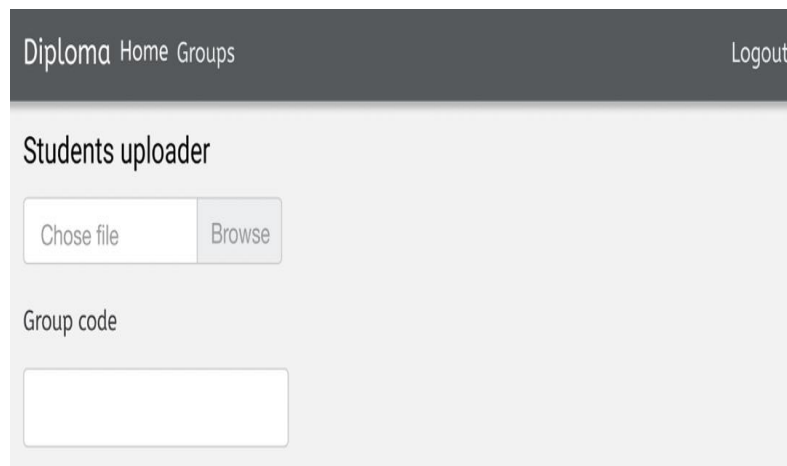


Рисунок 3.4 – Реєстрації груп.

Також можемо бачити кастомне розширення темплейту адмінки (розширення файлу верстки) – панель реєстрації груп (3.4). Доступне поле для завантаження групових файлів студентів для реєстрації їх в базі даних та можливість введення коду групи (3.5).



Рисунок 3.5 – Панель додавання таблиць груп.

Інтерфейс для створення таблиці предмету. Викладач обирає із доступних груп та вводить назву таблиці (3.6).

#	Table name	Group code	Enter	Edit	Remove
1	IK-91 Prog	IK-91	Enter	Edit	Remove
2	IK-91 ASM	IK-91	Enter	Edit	Remove
3	IT-82 OOP	IT-82	Enter	Edit	Remove

Рисунок 3.6 – Доступи і редагування таблиць груп.

Інтерфейс таблиць груп. Викладач може навігуватися по них та редагувати їх.

3.4. Розробка алгоритму роботи чат-бота для сповіщень користувачів

Після внесення списку групи в сервіс та запису студентів в базу даних програма повертає файл зі списком логінів та паролів для кожного студента. Досить просто роздати ці дані студентам, вони у свою чергу введуть ці дані в чат-бот і підпишуть себе на розсилку. При зміні Google таблиці на рядку напроти певного студента, по сумісництву користувача чат-бота, всередині гугл сервісів буде виконано метод `on edit`.

```
function onEdit(e) {
  return handleEdits(e.range);
}
function handleEdits(r) {
  s = r.getSheet();
  return s;
}
```

Таким чином отримуємо поточні зміни й можемо надіслати їх на сторону нашого сервісу за допомогою пушів гугла. Отримавши на вебхук сервісу запит від гугла, нам залишається знайти даного користувача в таблиці юзерів, його Telegram ID та надіслати йому повідомлення про зміну оцінки.

Запуск вебхук сервера Telegram-бота для отримання сповіщень від нових подій сервісу, а також нотифікацій про дії користувачів в Telegram від серверів телеграму.

```
server = flask.Flask(__name__)
@server.route('/') + TOKEN, methods=['POST'])
def get_message():
```

```

bot.process_new_updates([types.Update.de_json(flask.request.stream.read().decode(
("utf-8")))]))
    return "!", 200
@server.route('/', methods=["GET"])
def index():
    bot.remove_webhook()
    bot.set_webhook(url="https://".format(APP_NAME, TOKEN))
    return "Hello!", 200
if __name__ == "__main__":
    server.run(host="0.0.0.0", port=int(os.environ.get('PORT', 5000)))

```

Для нотифікацій користувачів Telegram, а саме студентів, про нові події буде використовуватися бродкаст повідомлень через чат-бот за допомогою методу надсилення повідомлень. Отримавши список користувачів, алгоритм кожному надішле повідомлення на його унікальний чат ID.

```
bot.send_message(message.chat.id, message.text)
```

3.5. Використання застосунку

Алгоритм використання вебсервісу викладачем:

Рисунок 3.7 – Створення таблиць груп.

Для початку роботи з групою викладач заходить в інтерфейс створення таблиць груп і створює таблицю-журнал для групи, назвавши її дисципліною (рис. 3.7)

Diploma Home Groups						Logout
Group tables						
#	Table name	Group code	Enter	Edit	Remove	
1	IK-91 Prog	IK-91	<button>Enter</button>	<button>Edit</button>	<button>Remove</button>	
2	IK-91 ASM	IK-91	<button>Enter</button>	<button>Edit</button>	<button>Remove</button>	
3	IT-82 OOP	IT-82	<button>Enter</button>	<button>Edit</button>	<button>Remove</button>	

Рисунок 3.8 – Інтерфейси доступів і редагування таблиць груп.

Після створення викладач може керувати цими таблицями та перейти безпосередньо до їх заповнення, натиснувши на посилання (рис. 3.8). Також він може здійснити розсилку всім студентам певної групи з інформацією релевантною для його предмету й сповістити таким чином їх про оновлення даних, або події, що плануються, наприклад контрольні роботи. Оскільки списки груп підтягують дані про студентів, то будь-яка дія в рядку навпроти ПІБ студента буде надсилатися йому в телеграм.

Підключення чат-бота студентом для отримання сповіщень про нові оцінки:

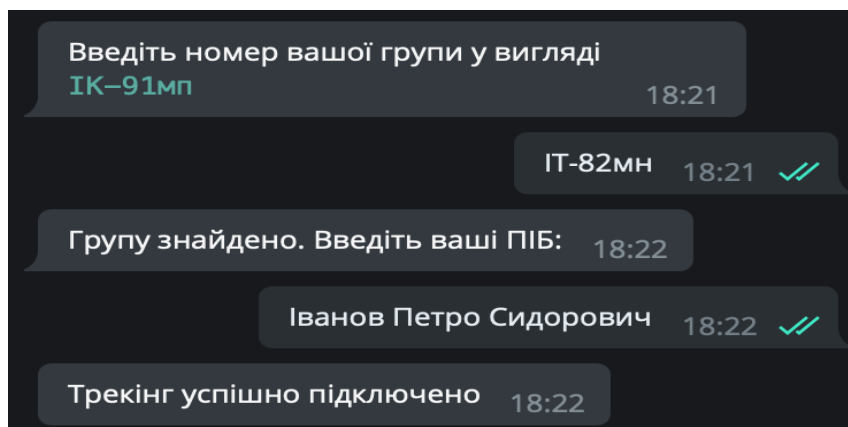


Рисунок 3.9 – Реєстрація в чат-боті для трекінгу оцінок.

Підключення чат-бота для отримання оцінок відбувається за простим алгоритмом. Студент знаходить чат-бот в телеграмі за посиланням, вводить шифр своєї групи та ПІБ. Якщо його знайдено в базі даних – йому надається доступ в чат-боті на отримання інформації та сповіщення про свою успішність (рис. 3.9). Оскільки списки груп мають коректні дані про студентів, то будь-яка дія в рядку навпроти ПІБ студента буде надсилатися йому в телеграм. В даному випадку на скріншоті нижче можемо бачити таке сповіщення (рис. 3.10).

Можемо бачити актуальну структуру сповіщення про подію предмету. Повідомлення містить назву предмету, дату події та її опис, взятий з першого рядка в таблиці, значення комірки, в даному випадку – оцінки студента й дані про викладача, який здійснив дану подію – поставив оцінку.

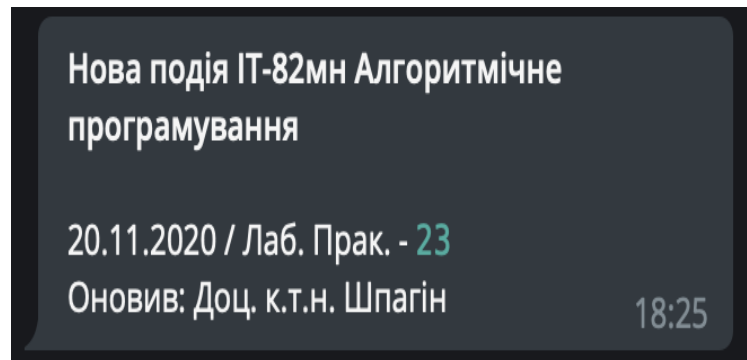


Рисунок 3.10 – Сповіщення про нову оцінку.

Висновки до розділу

Розроблено вебзастосунок для ведення обліку успішності студентів з можливістю завантаження списків груп, підписки на оновлення даних про успішність за допомогою телеграм чат-бота. Софт вирішує проблему своєчасного сповіщення студентів про поточні успіхи та навчальні події. Також закладено можливість зробити розсилку студентам певних груп, фільтруючи за кафедрою, чи взагалі усім користувачам чат-бота, наприклад з метою нагадування про атестацію.

4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ

4.1. Опис ідеї проєкту

Таблиця 4.1. Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створити вебсервіс для здійснення управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.	Використання викладачами для здійснення контролю за навчальними досягненнями студентів; як спосіб отримання студентами результатів власних навчальних досягнень; метод інформування про навчальні події	Здійснення контролю й оцінювання навчальних досягнень у дійсному часі; студенти можуть дистанційно стежити за власними досягненнями й новинами закладу, в якому навчаються; економія часу обох сторін; ефективний метод під час дистанційного навчання

Таблиця 4.2. Опис ідеї стартап-проєкту

Техніко-економічні характеристики ідеї	Продукція конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
	Мій проєкт	Конкурент 1	Конкурент 2	Конкурент 3			
Використання простих і зрозумілих технологій при розробці сервісу(Python,Django, Telegram).	Вебсервіс для управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.	Місцевий проєкт	Місцевий проєкт	Місцевий проєкт	Необхідність постійного оновлення актуальної інформації для ефективного функціонування вебсервісу; для отримання, обміном інформацією необхідний доступ до Інтернет-мережі.	Для користування достатньо мати комп'ютер, телефон або планшетний пристрій.	Можливість здійснювати спілкування із студентами дистанційно; зручність і доступність для користувачів; чат-бот, який здійснює інформування про зміни в оцінюванні.

Закінчення таблиці 4.2

Наявність bot-сповіщень для миттєвого отримання даних всіма користувачами.					Відсутність друкованого формату інформації.		модуль StudentUploader, який вивантажує студентів груп у базу даних з Excel таблиці.
Розмежування ролей та обліку всіх користувачів в проекті за допомогою стандартних моделей Джанго (таблиці викладачів, кафедр, студентів, предметів, груп)							
Можливість вивантаження списку студентів у базу даних							
Сервіс, що дозволяє згенерувати паролі для доступу кожному користувачеві							
Наявність засобів зміни гугл таблиці на рядку й надсилання поточних змін за допомогою телеграм ID студента							

4.2. Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Створення веб-сервісу для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.	Мова програмування Python фреймворк для веброзробки Django база даних PostgreSQL Python обгортка навколо API Google Sheets месенджер для інтеграції сповіщень. засобом чат-бота – Telegram.	Технологія наявна.	Технологія доступна й зручна Проблем у користуванні не передбачається.

Висновок: технологічна реалізація продукту – можлива, вибрана технологія №1

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4. Попередня характеристика потенційного ринку

	Показники стану ринку	Характеристика
	Кількість головних гравців, од	2, викладач, студент
	Загальний обсяг продаж, грн./ум.од	
	Динаміка ринку	Зростає
	Наявність обмежень для входу	Вхід обмежений тим, у кого відсутній пароль доступу до сервісу
	Специфічні вимоги до стандартизації та сертифікації ¹	Відсутні
	Середня норма рентабельності в галузі або по ринку, %	

Висновок: враховуючи кількість головних гравців на ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму

рентабельності, можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Створення сервісу для оцінювання навчальних досягнень студентів, контролю за виконанням навчального плану.	Викладачі як особи, котрі здійснюють контроль за успішністю студентів, студенти, котрі отримують інформацію про навчальні досягнення.	Більші вимоги до викладачів. Вони ведуть облік успішності студентів, заповнюють дані, інформують про новини.	Вебсервіс повинен бути зручним і зрозумілим у користуванні. Швидке завантаження даних.
2	Створення платформи, ефективної у період дистанційного навчання.			
3	Заощадження часу обох сторін (викладач-студент).			

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів, котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів
3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника.
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями, що відсутні у компанії конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників.
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи кешування даних.
4	Грошова винагорода за рекламу	При достатньому попиту на систему кешування даних можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвитку продукту та оплати заробітної плати працівникам	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проєкт задля його подальшого розвитку.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим замінником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками.	Розробка продукту з характеристиками, які покривають сфери вживання що не покривають інші товари-замінники; Кореляція цін у відповідності до товарів замінників; Різні типи ліцензій.

Закінчення таблиці 4.8

2	Рівень конкурентної боротьби: Світовий	Всі продукти замітники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників.	Вихід на ринок збуту продукту з клієнто-необхідною функціональністю; Налагодження маркетингу на основних Інтернет ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту.
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки ІТ додатків \ продуктів.	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки.
4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки.
5	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках.	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
6	За інтенсивністю: Марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників.	Впровадження власної назви та власного знаку.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

С к л а д о ві а н а лі з у	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Відсутні.	Потенційним конкурентами можуть стати розробники аналогічного вебсервісу.	Розробники пакету IDE PyCharm, за допомогою якого створений продукт можлива подальша реалізація оновлення.	Клієнти можуть знайти інший застосунок або не користуватись ним у разі відсутності оновлень, як інформаційного, читак і технічного характеру.	Товарами-замінниками є застосунок "Журнал викладача", "Мобільний журнал", електронний кампус НТУУ "КПІ".

Закінчення таблиці 4.9

В и с н о в к и	Боротьба з прямими конкурентами відсутня.	Є вільний доступ до ринку продажу за короткий проміжок часу.	Дії постачальників можуть ускладнити умови на ринку, проте не диктують їх.	Клієнти не мають вирішального впливу на ринкові умови, але в разі незадоволення своїх потреб можуть знайти аналогічний продукт.	Товари-замінники не здійснюють перешкод для функціонування на ринку.
--------------------------------------	---	--	--	---	--

Проаналізувавши можливості роботи на ринку, з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, то кожен з них має свою специфічну сферу використання та позитивні й негативні сторони щодо рішення певних типів задач, робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал, який відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний і достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування
Логіка сповіщень – bot	Користувач може взаємодіяти з ботом для миттєвого інформування.
Паролі для доступу	Кожен студент отримує пароль для доступу до сервісу й підписки на розсилку.
Інформування про зміну оцінки	Студенти отримують інформацію про поточні зміни в оцінюванні.
Зручний і зрозумілий інтерфейс	Технології, обрані для реалізації застосунку, не викликають труднощів при користуванні.

**Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін системи
кешування малозмінних даних**

Фактор конкурентоспроможності	Кількість балів (1-10)			
	«Журнал викладача»	«Мобільний журнал»	Електронний кампус НТУУ «КПІ»	Веб-сервіс для обліку успішності
Інтерфейс	9	5	4	8
Інформативність	9	7	5	7
Інноваційний характер	8	6	7	9
Маркетинг	9	6	5	8
Сума балів	35	24	21	32

Висновок – застосунок «Журнал викладача» має найвищу кількість балів

Таблиця 4.12. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – Наявність Telegram Bot для швидкого отримання інформації; – Зручний та зрозумілий інтерфейс; – Інформування про зміну поточного бала; – Особистий пароль для доступу до даних, що надається кожному користувачеві; – Інноваційна складова застосунку; – Вивантаження списку груп студентів у базу даних з Excel таблиці. – Для запуску програми не потрібно значних фінансових витрат. 	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Потреба в Інтернет-мережі для обміну й отримання інформації; – вузькоспеціалізованість -платформа призначена для викладачів і студентів; – Для функціонування необхідне докладання зусиль й постійне оновлення.
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Вільний доступ до інформації в будь-який час; – Використання сучасних технологій при створенні (Telegram Bot); – Можлива реалізація вебсервісу в навчальному процесі НТУУ "КПІ"; – Використання застосунку в період дистанційного навчання; – Подальше оновлення версії вебсервісу, доступної ширшій цільовій аудиторії. 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – Можлива поява аналогічних вебсервісів, що може спричинити перехід цільової аудиторії на більш зручні застосунки; – Зниження попиту у випадку відсутності інформаційних і технічних оновлень.

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проєкту

Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс – наявний	2-3 місяці
Реклама	Залучення власних коштів для реклами товару	1-2 місяці
Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс – наявний	2-3 тижні
Презентація товару на хакатонах й інших ІТ-заходах	Ресурс – час та гроші для участі, наявні	1-3 місяці

4.4. Розроблення ринкової стратегії проєкту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Викладачі	Низька	Низький	Низька	Висока
Студенти	Висока	Високий	Низька	Висока

Які цільові групи обрано: викладачів і студентів

Відповідно до проведеного аналізу можна зробити висновок, що доцільною цільовою групою для розповсюдження даного програмного продукту є працівники ІТ-сфери, ІТ-компанії та будь-які підприємства, котрі використовують програмні продукти, побудовані мовою програмування Java, і користуються реляційними базами даних. Відповідно до схеми охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для підприємств, ІТ-працівників та ІТ-компаній надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії).

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності, що відсутня у товарів-замінників, підтримка клієнтів.	Проведення реклами, освітлення унікальної функціональності через Інтернет-ресурси та інші канали, контакт напряму з споживачами; формування лояльності і прихильності споживачів.	Зниження ступеню замінності товару; Прихильність клієнтів; Відмітні властивості товару; Відмітні характеристики товару.	Стратегія диференціації.

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проєкт «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари замінники не мають деякого необхідного функціоналу.	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх.	Компанія частково копіює характеристики товару конкурента, основна ціль компанії розробка нового унікального функціоналу, з підтримкою основного функціоналу конкурентів.	Стратегія заняття конкурентної ніші.

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару з точки аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції даного стартап-проєкту	Вибір асоціацій, які сформувати експерту позицію власного проєкту
1	Простий і зрозумілий інтерфейс	Стратегія диференціації	Доступність	Простота й зручність
2	Інформаційне й технічне оновлення продукту	Стратегія диференціації	Модернізація	Інформативність
3	Чітка й структурована побудова функцій (списків груп, функцій, оцінок, рейтингів)	Стратегія диференціації	Високоякісність	Чіткість

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку стратегію диференціації, а як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Простий інтерфейс	Правильна й чітка розробка застосунку	На відміну від вебсервісів-конкурентів, застосунок створений за допомогою універсального стеку технологій (Python, Django, PostgreSQL, Python-обгортка, Telegram).
2	Постійне оновлення	Оновлення можуть здійснювати користувачі вебсервісу (студенти й викладачі)	Можливе подальше оновлення зробить вебсервіс однією із основних платформ для обліку успішності студентів (на відміну від застарілого інтерфейсу електронного кампуса КПП ім. Ігоря Сікорського).
3	Інформативність	Студенти отримують інформацію про поточні бали, корекцію оцінок чи новини університету	Отримання інформації за допомогою чат-бота.
4	Доступність	Застосунком можна користуватись з будь-якого пристрою (телефон, комп'ютер, планшет).	Вебсервіс доступний із різних пристроїв (на відміну від «Мобільного журналу»).

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система кешування малозмінних даних мовою програмування Java		
II.Товар у реальному використанні	Властивості / характеристики	М / Нм	Вр / Тх / Тл /Е / Оп
	Зручний і зрозумілий інтерфейс	М	Тх
	Повідомлення про зміну оцінки	Нм	Е
	Доступний для використання	М	Е
	Сучасні технології інформування (Telegram)	Нм	Тх
	Безпечний для користувача (студенти отримують пароль і логін для доступу)	М	Е
	Наявність чіткої структури (таблиці викладачів, предметів, студентів, оцінок)	М	Е
	Якість: відповідає вимогам якості Пакування: відсутнє Марка: вебсервіс для здійснення реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.		
III. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент			

Таблиця 4.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Високий	Середній	Низький	Верхня та нижня межі встановлення ціни на товар/послугу

Таблиця 4.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнту для отримання доступу необхідно встановити застосунок і авторизуватись.	Розробити чіткий та зрозумілий інтерфейс. Встановити зворотній зв'язок із клієнтом.	Кількість користувачів, що передають інформацію про вебсервіс один одному.	Збут через учасників навчального процесу.

Таблиця 4.22. Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Викладачі та студенти як головні цільові клієнти можуть користуватись вебсервісом при умові постійного оновлення інформації й бажанні використовува ти електронні сервіси для обліку успішності.	Офіційн ий сайт університету, згадки в періодичних виданнях, соціальні мережі, реклама, рекомендації.	Зручність, структурованість, доступність, простота, безпечність, чіткість, модернізованість.	Переконат и потенційних клієнтів, що використання застосунку значно полегшить роботу й зекономить час.	Навести клієнту переконливі аргументи переваг застосунку.

Відповідно було створено ринкову (маркетингову) програму, до якої входить визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки до розділу

У четвертому розділі описано стратегії та підходи з розроблення стартап-проєкту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано, що існує можливість ринкової комерціалізації проєкту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан

конкуренції та конкурентоспроможність проєкту було встановлено, що він є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проєкту й доведено доцільність подальшої імплементації проєкту.

ВИСНОВКИ

Введення автоматизації сповіщень студентів про навчальні досягнення і рейтинги є сьогодні невід'ємною частиною для успішного розвитку освітньої галузі. Зважаючи на пандемію 2020 року, цей процес потребує вдосконалення і подальшого розвитку.

На даний час не існує централізованої системи, здатної до синхронізованої роботи зі своєчасним оновленням. Вже наявні застосунки ("Журнал викладача", "Мобільний журнал", "Електронний Кампус КПІ") не повністю відповідають вимогам користувачів. Результатом вирішення цього питання стала розробка вебсервісу для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних.

Застосунок є платформою, що дозволяє викладачам створювати таблицю-журнал груп із певної дисципліни, керувати нею, здійснювати обмін інформацією (повідомлення про оновлення даних чи інші події), студентам дає можливість отримувати новини навчального закладу й контролювати власні досягнення при вивченні предмета.

Перевагою даного вебсервісу є наявність чат-бота в Telegram, за допомогою якого здійснюється інформування студента про дії в рядку напроти його ПІБ, у тому числі й повідомлення про успішність. Так студент знаходить чат-бот в Telegram за посиланням, вводить шифр своєї групи й ПІБ, після чого отримує доступ до даних щодо успішності та актуальних новин університету. Повідомлення містить назву предмета, дату події, її опис, значення комірки й дані про викладача, який здійснив подію.

Створення такого вебсервісу значно спрощує роботу викладача, а студентам дозволяє контролювати власні досягнення із дисципліни. Застосунок створений із метою задоволення потреб споживачів, дозволяє зекономити час, є ефективним у період дистанційного навчання. З його допомогою можна здійснювати контроль успішності в дійсному часі. Зручний та зрозумілий

інтерфейс є значною перевагою і виділяє вебсервіс для обліку успішності студентів з-посеред інших подібних додатків.

Проведене маркетингове дослідження засвідчило наявність позитивних характеристик і усіх необхідних інструментів для виходу на ринок. Наявність попиту, динаміки, рентабельності роботи, потенційних клієнтів, бар'єрів входження, стану конкуренції засвідчили перспективність і актуальність проекту.

ПЕРЕЛІК ПОСИЛАНЬ

1. О. І. Лещенко, Г. С. Бондаренко, АНАЛІЗ ПРОБЛЕМ РОЗРОБКИ РОЗПОДІЛЕНОЇ СИСТЕМИ КОНТРОЛЮ ТА ОБЛІКУ Галелюка І.Б. Моделювання бездротових сенсорних мереж / І.Б. Галелюка // Комп'ютерні засоби, мережі та системи. – 2015. – No 14. – С. 141 – 150.
2. В.І.Голіков, О.Ю.Євганов, М.В.Фатєєв, В.Д.Чайка. – Миколаїв: УДМТУ, 2004. – 184 с. Організація інформаційних систем в управлінні : навчальний
3. Роберт Никонс. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 3-е изд. – СПб.: Питер, 2015. – 688с. Котеров Д.В. PHP7 / Д.В. Котеров, И.В. Симдянов. – СПб.: БХВ-Петербург, 2016. – 1088с.
4. Васильєв В.М. СИСТЕМА ОБЛІКУ ВІДВІДУВАНOSTІ Й УСПІШНОСТІ СТУДЕНТІВ – ЯК ЗАСІБ МОНІТОРИНГУ РІВНЯ ЗНАНЬ У СТУДЕНТІВ
5. Круглик В.С. Сучасні підходи до використання інформаційно-комунікаційних технологій у навчанні // Херсон: Вид. ХДУ. – 2008. – С. 114- 119.
6. Положення про електронний журнал обліку навчальної роботи студентів академічної групи [Електронний ресурс]. – Режим доступу: <http://kneu.edu.ua/userfiles/center/polojennya2.doc>.
7. Кірвас В.А. Автоматизований моніторинг та облік навчальної діяльності студентів при кредитно-модульній системі підготовки фахівців / В.А. Кірвас // Наука і соціальні проблеми суспільства: освіта, культура, духовність. Матеріали Міжнародної науково-практичної конференції 20-21 травня 2008 р. – Харківський національний педагогічний університет імені Г.С. Сковороди. – С. 45-49.
8. Ярошинський В.М. Методика підготовки тестів успішності при викладанні дисциплін «Бухгалтерський облік у бюджетних установах» / В.М. Ярошинський Я.М. Рудик [Електронний ресурс]. – Режим доступу: URL: <http://elibrary.nubip.edu.ua/10505/1/11rym.pdf>.

ДОДАТКИ

ДОДАТОК А
Відомість проекту

з/п	Формат	Позначення	Найменування	Кільк ість листів	рим ітка
	A4		Завдання на дипломний проект		
	A4		Пояснювальна записка		
	A4		Додаток А. Відомість проекту		
	A4		Додаток Б. Результат перевірки на співпадіння		
	A3		Додаток В. схема бази даних		
	A3		Додаток Г – архітектура вебсервісу		
	A3		Додаток Д – опис моделей класів ORM		
	A3		, Додаток Ж – 6 плакатів		

ДОДАТОК Б

Результат перевірки на співпадіння



Ім'я користувача:
Лісовиченко Олег Іванович

ID перевірки:
1005344355

Дата перевірки:
03.12.2020 12:16:51 EET

Тип перевірки:
Doc vs Library

Дата звіту:
03.12.2020 12:18:23 EET

ID користувача:
76913

Назва документа: Співпадіння_Юраш_3

Кількість сторінок: 48 Кількість слів: 7407 Кількість символів: 61062 Розмір файлу: 2.67 MB ID файлу: 1005636735

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.84%
Схожість

Найбільша схожість: 0.41% з джерелом з Бібліотеки (ID файлу: 1003082174)

Пошук збігів з Інтернетом не проводився

1.84% Джерела з Бібліотеки

133

Сторінка 50

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

12
сторінок

ДОДАТОК В
Схема бази даних

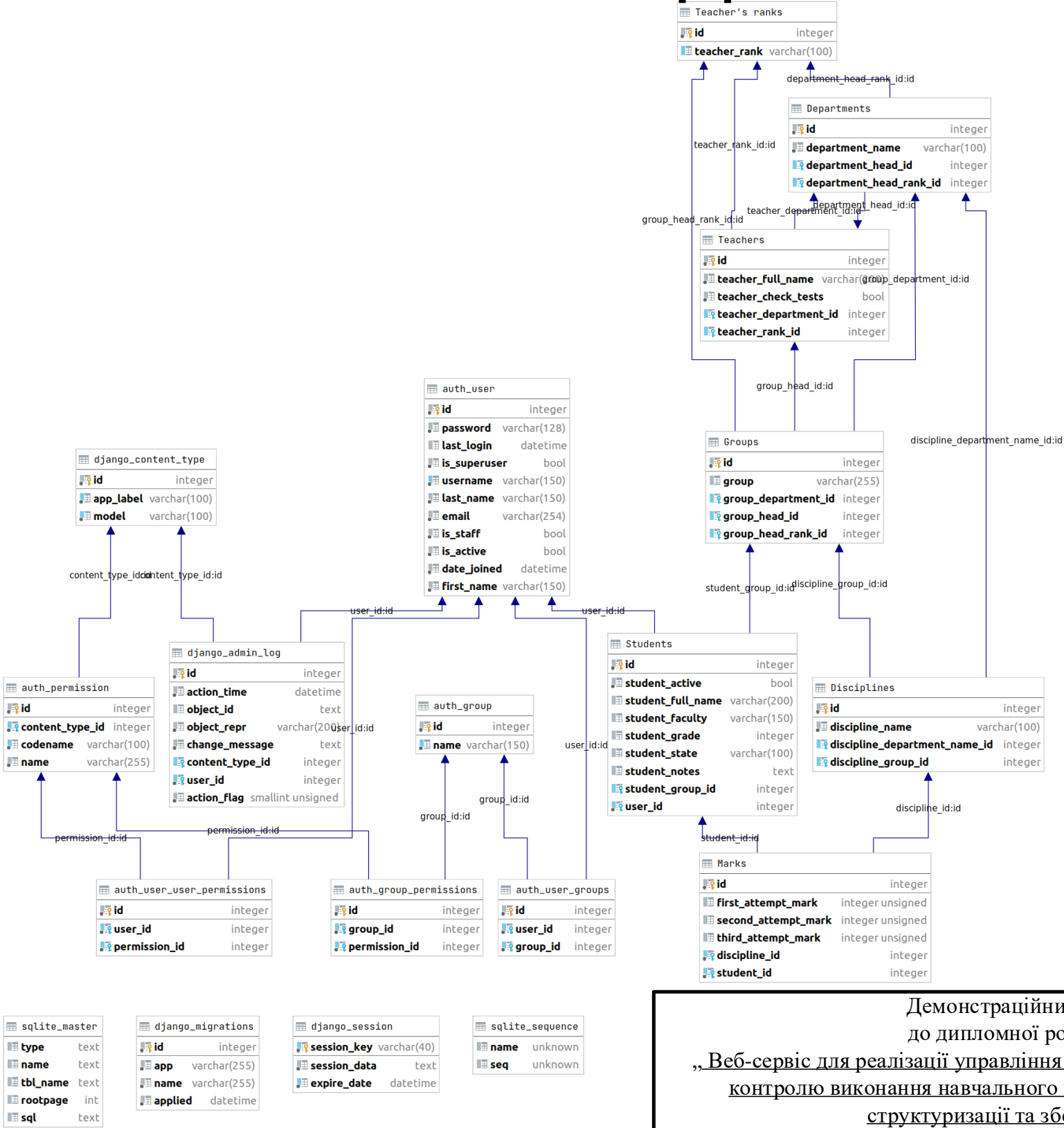
ДОДАТОК Г

Архітектура вебсервісу

ДОДАТОК Д
Опис моделей класів ORM

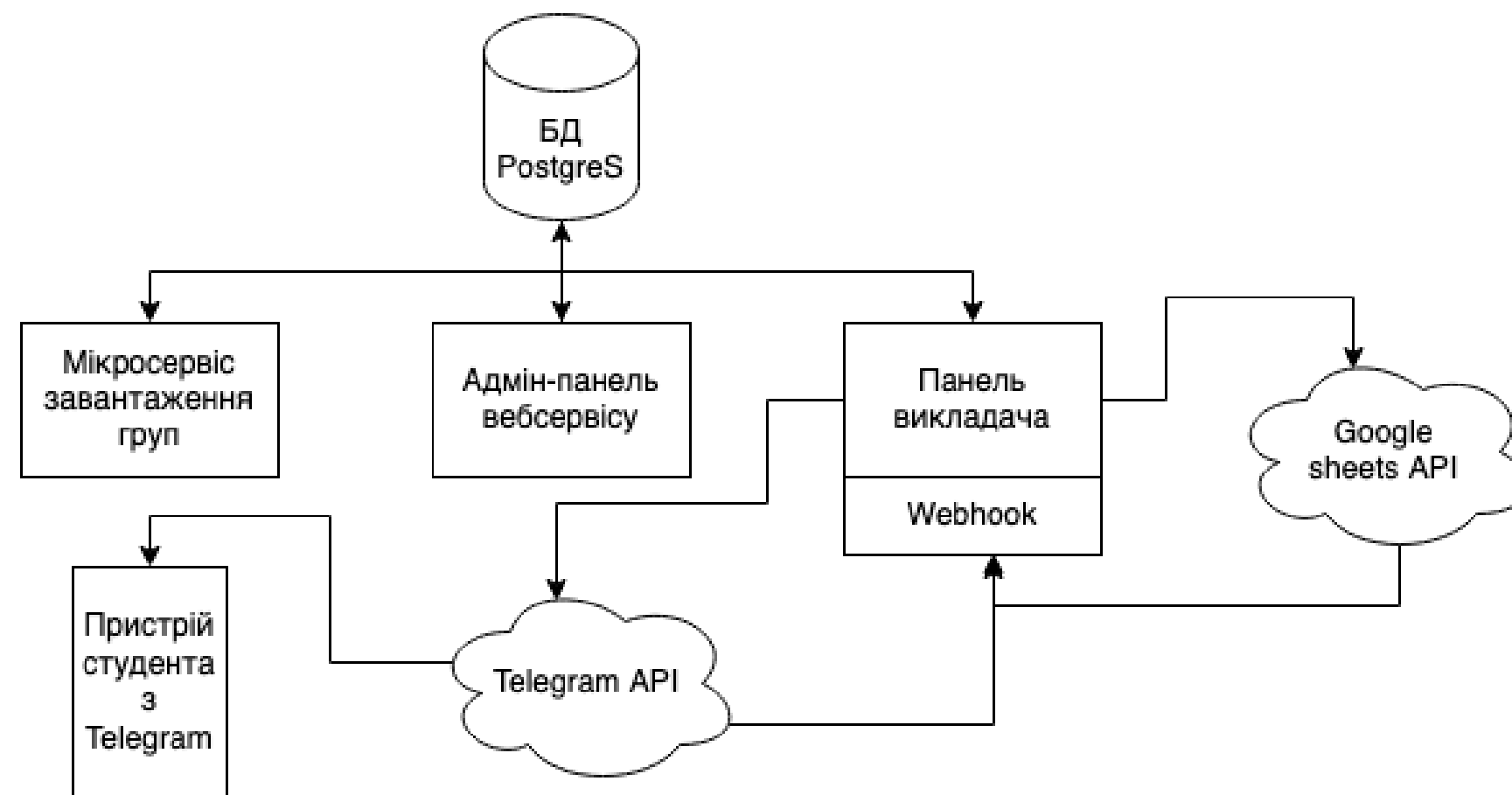
ДОДАТОК Ж
Плакати

Схема бази даних



Демонстраційний плакат № 1
до дипломної роботи на тему
„Веб-сервіс для реалізації управління інтерактивною системою обліку та
контролю виконання навчального плану з інтеграцією сервісу для
структуризації та збереження даних ”
Розробив: студент групи ІК-91мп Юраш А.П.
Керівник: доц. кафедри ТК ФІОТ, к. т. н., доцент Лісовиченко О.І.

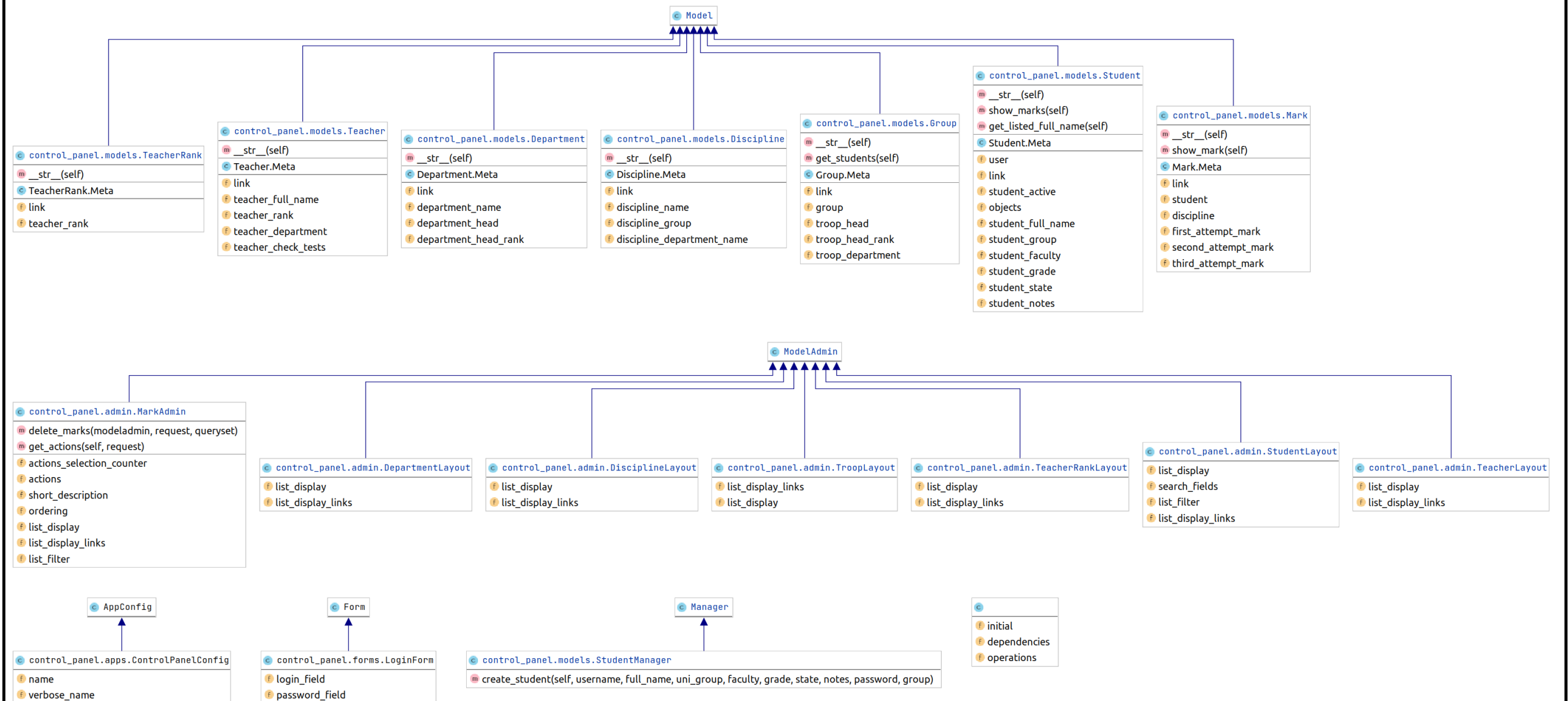
Архітектура веб-сервісу



Демонстраційний плакат № 2
до дипломної роботи на тему
„Веб-сервіс для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних”

Розробив: студент групи ІК-91мп Юраш А.П.
Керівник: доц. кафедри ТК ФІОТ, к. т. н., доцент Лісовиченко О.І.

Опис моделей класів ORM

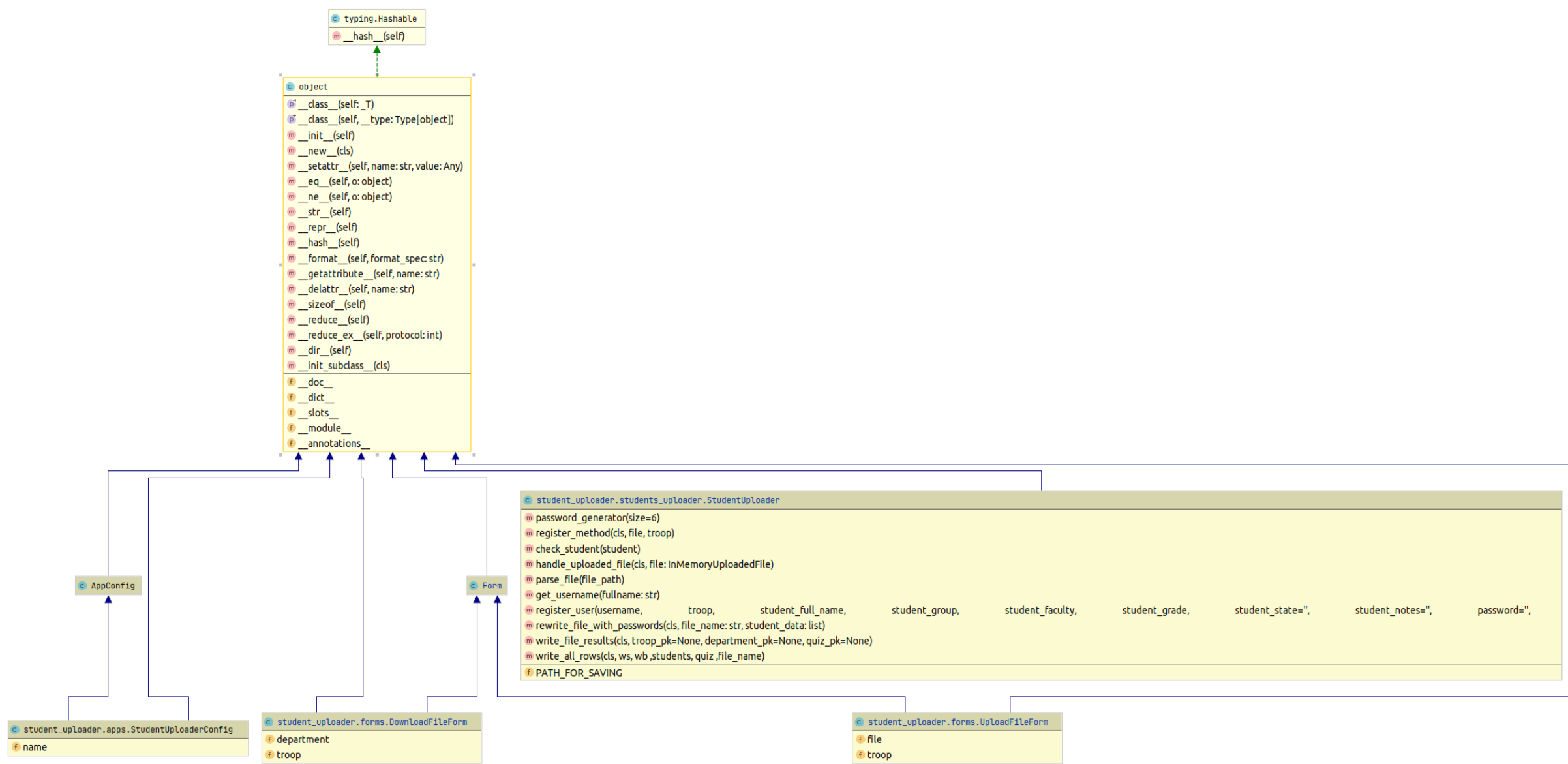


v

Демонстраційний плакат № 3
до дипломної роботи на тему
„Веб-сервіс для реалізації управління інтерактивною системою обліку та
контролю виконання навчального плану з інтеграцією сервісу для
структуризації та збереження даних ”

Розробив: студент групи ІК-91мп Юраш А.П.
Керівник: доц. кафедри ТК ФІОТ, к. т. н., доцент Лісовиченко О.І.

Схема модуля student_uploader



Демонстраційний плакат № 4
до дипломної роботи на тему
„Веб-сервіс для реалізації управління інтерактивною системою обліку та
контролю виконання навчального плану з інтеграцією сервісу для
структуризації та збереження даних ”
Розробив: студент групи ІК-91мп Юраш А.П.
Керівник: доц. кафедри ТК ФІОТ, к. т. н., доцент Лісовиченко О.І.

Приклади роботи веб-сервіса

Адміністрування сайта

АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ

Групи

+ Додати

Змінити

Користувачі

+ Додати

Змінити

КЛЮЧОВІ ТАБЛИЦІ

Викладачі

+ Додати

Змінити

Таблиці оцінок

+ Додати

Змінити

Оцінки

+ Додати

Змінити

Предмети

+ Додати

Змінити

Студенти

+ Додати

Змінити

Панель реєстрації груп

[На панель](#)

Diploma Home Groups Logout

Students uploader

Chose file

Browse

Group code

Choose group: Enter table name

Groups

IK-91

IT-82

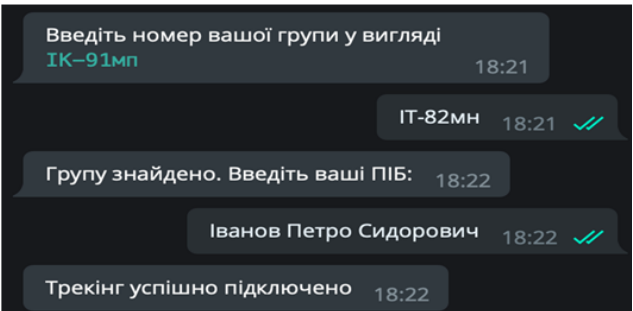
IK-91 Programming C

Create

Демонстраційний плакат № 5
до дипломної роботи на тему
„Веб-сервіс для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних”
Розробив: студент групи ІК-91мп Юраш А.П.
Керівник: доц. кафедри ТК ФІОТ, к. т. н., доцент Лісовиченко О.І.

Приклади роботи веб-сервіса – продовження

#	Table name	Group code	Enter	Edit	Remove
1	IK-91 Prog	IK-91	Enter	Edit	Remove
2	IK-91 ASM	IK-91	Enter	Edit	Remove
3	IT-82 OOP	IT-82	Enter	Edit	Remove



Diploma Home Groups Logout

Group tables

#	Table name	Group code	Enter	Edit	Remove
1	IK-91 Prog	IK-91	Enter	Edit	Remove
2	IK-91 ASM	IK-91	Enter	Edit	Remove
3	IT-82 OOP	IT-82	Enter	Edit	Remove

Diploma Home Groups Logout

Choose group:

Enter table name

Groups

IK-91

IT-82

IK-91 Programming C

Create

Демонстраційний плакат № 6
до дипломної роботи на тему
„Веб-сервіс для реалізації управління інтерактивною системою обліку та контролю виконання навчального плану з інтеграцією сервісу для структуризації та збереження даних”
Розробив: студент групи ІК-91мп Юраш А.П.
Керівник: доц. кафедри ТК ФІОТ, к. т. н., доцент Лісовиченко О.І.